STEREO VISION: A COMPARISON OF SYNTHETIC IMAGERY VS REAL WORLD IMAGERY FOR THE AUTOMATED AERIAL REFUELING PROBLEM

THESIS

Nicholas J. Seydel, 2nd LT, USAF

AFIT-ENG-MS-18-M-059

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-18-M-059

STEREO VISION: A COMPARISON OF SYNTHETIC IMAGERY VS REAL WORLD IMAGERY FOR THE AUTOMATED AERIAL REFUELING PROBLEM

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Science

Nicholas J. Seydel, B.S.C.E.

2nd LT, USAF

March 2018

AFIT-ENG-MS-18-M-059

STEREO VISION: A COMPARISON OF SYNTHETIC IMAGERY VS REAL WORLD IMAGERY FOR THE AUTOMATED AERIAL REFUELING PROBLEM

THESIS

Nicholas J. Seydel, B.S.C.E.
2nd LT, USAF

Committee Membership:

Dr. Scott Nykl
Chair

Dr. Robert Leishman
Member

Dr. Douglas Hodson
Member

AFIT-ENG-MS-18-M-059

Abstract

Missions using unmanned aerial vehicles have increased in the past decade. Currently, there is no way to refuel these aircraft. Accomplishing automated aerial refueling can be made possible using the stereo vision system on a tanker. Real world experiments for the automated aerial refueling problem are expensive and time consuming. Currently, simulations performed in a virtual world have shown promising results using computer vision. It is possible to use the virtual world as a substitute environment for the real world. This research compares the performance of stereo vision algorithms on synthetic and real world imagery.

Acknowledgements

Thank you to the Air Force Research Lab for sponsoring this research.

Nicholas J. Seydel

Table of Contents

Page

List of Figures

List of Tables

STEREO VISION: A COMPARISON OF SYNTHETIC IMAGERY VS REAL WORLD IMAGERY FOR THE AUTOMATED AERIAL REFUELING PROBLEM

I. Introduction

1.1 Overview/Background

The United States Air Force (USAF) has developed a strong dependence on Unmanned Aerial Vehicles (UAV). These aircraft are known for flying lengthy intelligence missions. More recently, UAVs have entered into combat with next generation Unmanned Combat Aerial Vehicles (UCAV). UCAVs extra weaponry does not allow for the same flight time endurance found in reconnaissance-oriented UAVs. This impediment makes it paramount that aerial refueling be made available to UCAVs, just as it is for modern combat aircraft. Current USAF refueling tankers require a boom operator seated in the rear pod of the tanker. This operator actuates the boom to conduct aerial refueling. To enable docking between a UCAV and a tanker, the process relies heavily on constant communication and precise movements of the receiving aircraft. This requirement is a major safety concern due to the latency that exists between a UAV operator and the UAV. Several seconds of latency between the command and execution create a dangerous environment for the UAV and the boom operator.

Modern tankers are replacing the rear pod with a stereo vision system. This system provides a digital video feed enabling the use of stereo vision algorithms. With the use of these algorithms, the tanker can potentially guide a UAV into the proper refueling position for a mid-flight refuel. Eventually, it may be possible to automate the refueling process to replace the boom operator, thus reducing the flight

crew. Motivated by this, the Air Force Institute of Technology (AFIT) is researching possible solutions for the Automated Aerial Refueling (AAR) problem.

The contribution of this work compares synthetic stereo imagery against real stereo imagery to evaluate the feasibility and efficiency of using 3D virtual worlds for AAR related development. Research at AFIT has produced a virtual environment capable of simulating stereo computer vision using synthetic imagery. Simulations in the virtual world estimate the 6 degrees of freedom (6DoF) of the receiver using the iterative closest point (ICP) algorithm. The performance of ICP must still be compared using real and synthetic imagery from the same experiments.

1.2 Problem Statement

Quantify and compare the performance of computer vision using real world imagery and synthetic imagery taken from a 3D virtual world.

1.3 Research Goals and Hypothesis

- Create a virtual environment nearly identical to a real world environment for re-creation of real world experiments
- Conduct experiments in the real world collecting truth data with less than 2mm error
- Replay truth data in the virtual environment
- Quantify the accuracy of ICP using computer vision on real world imagery
- Quantify the accuracy of ICP using computer vision on synthetic imagery
- Compare the error in position estimation for both environments
- Find trends in position estimation for both environments

- Show synthetic imagery can be used in lieu of real imagery without affecting the vision processing results

1.4 Approach

Experiments involve flying and tracking a quadcopter Parrot Bebop drone. These flights were recreated in the virtual world using a model of the quadcopter. The experiments captured by stereo vision cameras can be replicated in the 3D Virtual World and its corresponding synthetic imagery can be captured. Computer vision algorithms can then process the real and synthetic imagery. The accuracy and behavior of these estimations can be compared.

1.5 Assumptions and Limitations

The propellers of the quadcopter drone were removed from the aircraft. To achieve a mock top down view, the drone was connected to clear fishing line. This allowed the orientation to be perpendicular to the ground, displaying the top of the drone to the cameras. It was then "flown" perpendicular to the ground in several flight paths. These flights allowed the drone to cover most of the volume that the cameras could see. While not accurately scaled down, this volume mimics the volume of space that makes up the refueling envelope. The scaling is not one to one because the reduction in the cameras baseline still does not allow a scaled aircraft to fit in the real world environment used. However, the cameras used have the same 56 degree field of view as used in previous research. The drone distance from the cameras covers a scaled down representation of a receiver in the refueling envelope.

The position and orientation of the drone was captured and logged to replay experiments in the virtual world. These logs also serve as the truth data and are accurate within a 2mm margin. The virtual sensors follow the pinhole camera model

without lens distortion yet should produce the same quality image as the real world.

1.6 Research Contributions

- Quantifies the accuracy of computer vision using real world imagery
- Quantifies the accuracy of computer vision using synthetic imagery from a 3D virtual world
- Shows behavioral trends that are present in both environments

1.7 Terms

Throughout this research there are many terms that need to be defined in order to understand the foundation.

Real World

- Refers to the real, physical world. Experiments are captured and recorded using both truth positioning and stereo cameras.

Virtual World

- The AFTR BURNR Engine: This is a game engine which allows human interaction to navigate and and change perspectives. This is the environment where simulations are performed and the real world Experiments are re-created.

Vicon Chamber

- Real world chamber capable of producing truth positioning and orientation data with up to 2mm error. Also where the location for real world experiments are conducted.

Simulation

- The replaying of real world experiments in the virtual world.

Real Imagery

- Stereo images captured by real digital cameras.

Synthetic Imagery

- Stereo images captured by virtual sensors in the virtual world.

1.8 Thesis Overview

Chapter 2 examines the background of the AAR problem. Additionally, this chapter explains the fundamentals of stereo vision, computer vision, and the 3D virtual world. A literature review of previous and related work finish the chapter. Chapter 3 describes the methodology of the experiments. The description includes the computer vision pipeline, the real world equipment used, and the equivalent virtual environment. The chapter elaborates on the process of estimating the drone's position and the reference model's role. The experiments conducted are explained at the end. The analysis and visualization of these experiments is presented in Chapter 4. The final chapter provides a conclusion as well as possible future research relating to this topic.

II. Background

2.1 Automated Aerial Refueling (AAR)

Aerial refueling has been a capability that has allowed the USAF to increase its global reach. Aerial refueling consists of a tanker aircraft providing fuel to a receiver aircraft. There are two refueling methods, the probe and drogue method and the boom method as seen in Figure 1. This research aims to assist the USAF's method of refueling, the boom method, developed by Boeing [3]. The boom method uses a boom controlled by a human operator. The receiver aircraft must enter the refueling envelope, a 3D volume behind the tanker. Receiver positioning relative to the tanker must be very precise for the boom operator to guide the boom.

UAVs are used for important roles in the USAF. These roles include reconnaissance and more recently, combat. The UCAVs do not have the same range as reconnaissance UAVs. The shorter range of the UCAVs restrict their effectiveness on the battlefield. This restriction can be solved using AAR, a capability that is not yet possible. AAR is currently hindered by the latency between the drone operator and the UAV. The distance between the drone operator and the UAV can delay commands by several seconds. This latency makes aerial refueling too dangerous due to the precision and

(a) Boom Method [4] **(b) Probe and Drogue Method [5]**

Figure 1. Refueling Methods

timing that is required. Latency can be eliminated if the UAV can be automated with commands from the tanker. While the UAVs do have global positioning systems (GPS), this technology may not be accurate enough for AAR. Also, the tanker can potentially block the receiver's GPS when in the refueling envelope or denied in contested environments. The use of computer vision combined with data from inertial measurement units (IMUs) can aid in providing the receiver's relative six degrees-of-freedom (6DOF). This information is crucial to solve the AAR problem.

2.2 Computer Vision

Humans their visual cortex to understand the depth and orientation of objects. This capability can be achieved with computers using complex algorithms. Computer vision provides geometry, orientation, and other attributes of imagery [6]. Computer vision must use features in imagery such as corners, colors, and patterns [7] to mimic human-like capabilities. With multiple cameras and the right algorithms, computer vision can be used to track and estimate the range of objects [8]. These capabilities are critical for AAR.

2.3 Pinhole Camera Model

Images produced by the pinhole camera model can be used for computer vision algorithms. Figure 2 shows the pinhole camera model. The optical center, F_c, is the origin of the camera frame. An image plane exists a certain length from the optical center. This plane is parameterized by x and y. An object point P projects to an image point *(u,v)*. This image point lies on the virtual image plane. When P projects to *(u,v)*, the depth information is lost. This is because the *(x, y, z)* parameters are mapped to *(u,v)*. The principal point, represented by (c_x, c_y), is intersected by the depth axis, Z [1]. The focal length of the camera is the distance

between the optical center and the principal point. These parameters work together to accomplish accurate estimations of real world objects using computer visions. While the real world cameras use a lens, an object point P will still map to a pixel with an x and y coordinate. Calculating the same parameters for a real world camera is still necessary for stereo vision.

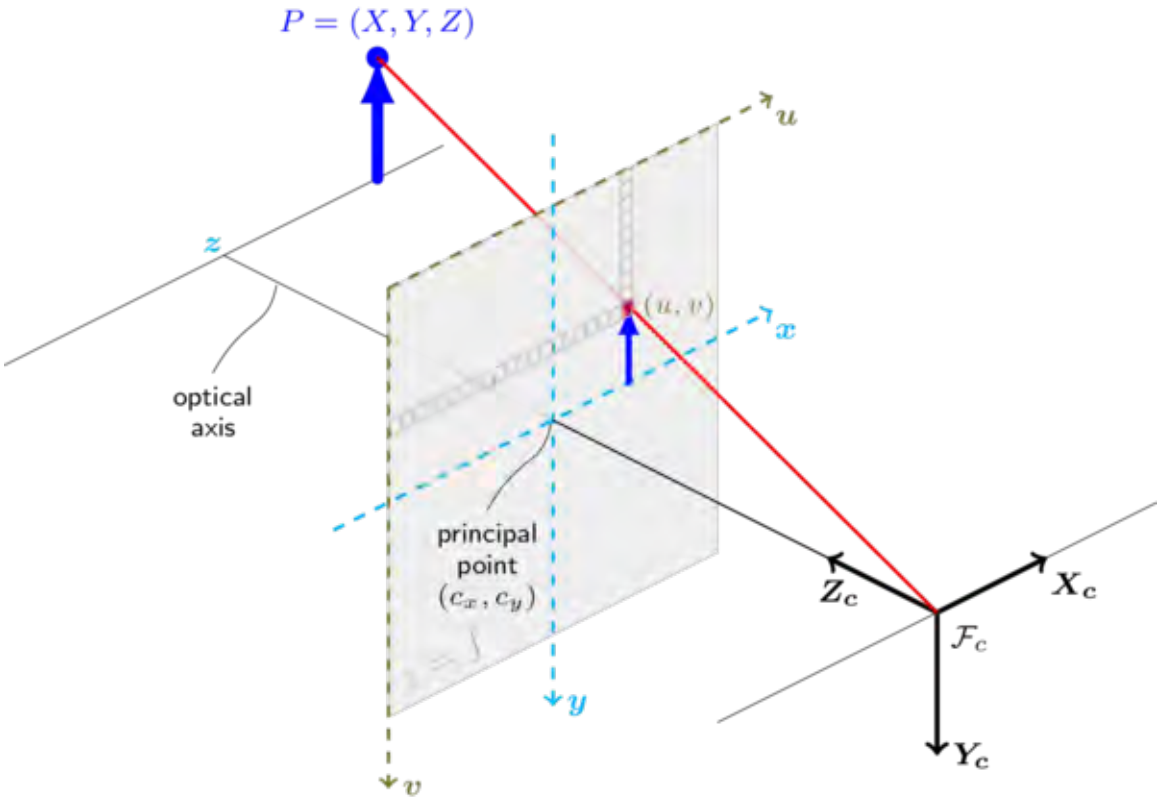


Figure 2. Pinhole Camera Model [1]

2.4 Stereo Computer Vision

Stereo Vision uses two cameras to mimic human-like depth perception. The depth can only be calculated when an object is in view of both cameras. This estimation is achieved using epipolar geometry [9]. The epipolar geometry of a stereo vision system can be seen in Figure 3. The object point, q, is in view of both the cameras. The cameras' optical centers are represented by O_1 and O_2. The baseline is the distance

between the two optical centers. The points where baseline passes through the image planes are the epipoles, e_1 and e_2. The object point, q, is portrayed on the image plane by the projections p_1 and p_2. All these components are used to create the epipolar plane. The epipolar lines are represented by l_1 and l_2. Epipolar geometry creates a relationship between the image point in one image with the epipolar line of the other image. This relationship can be condensed using an essential matrix. The essential matrix relates both the image points and the object point q. This matrix also relates the geometry between the two cameras. These points define the epipolar constraint, represented by $p_1^\top E p_2 = 0$ [9]. The epipolar constraint allows a matching pixel to be found on the corresponding epipolar line. Similarly, the fundamental matrix describes this relationship in more general terms. The fundamental matrix differs from the essential matrix in that it also includes the intrinsic properties of the cameras.

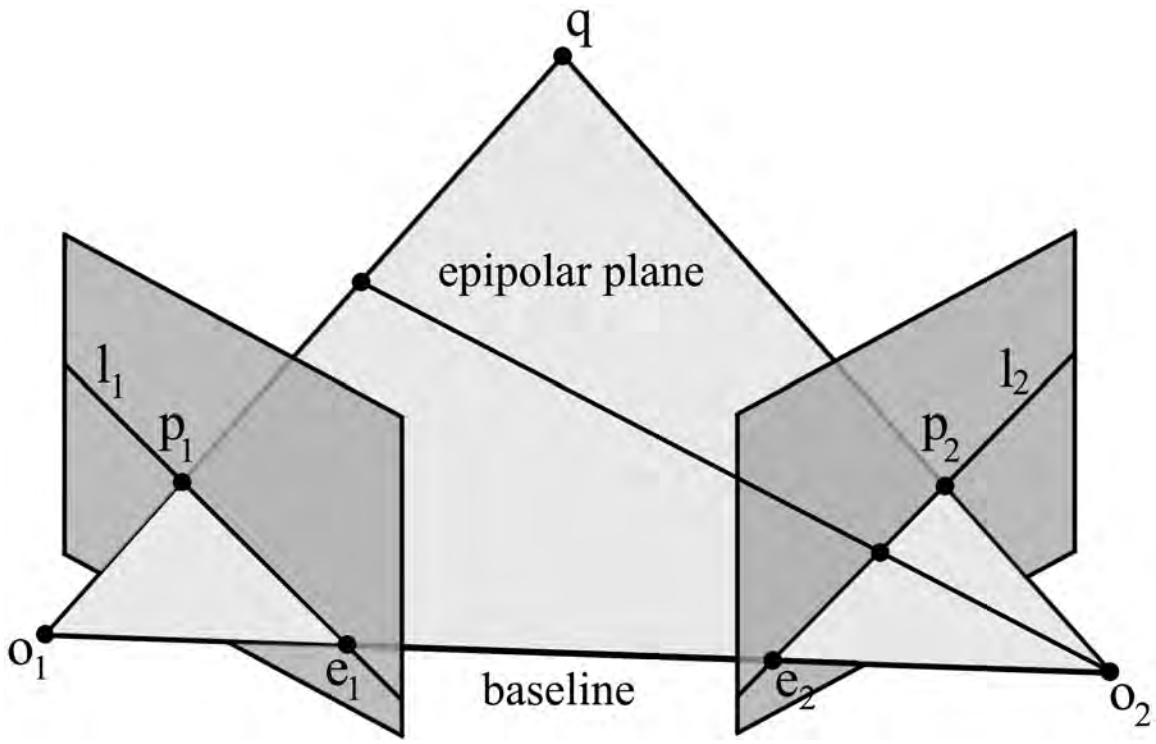


Figure 3. Stereo Vision Epipolar Geometry [2]

Epipolar geometry is essential for finding 3D coordinates. Given two images, it produces estimated depths of object points in terms of the primary camera's optical center. It is required that the images from the left and right camera are taken at the exact same time. Stereo vision requires knowledge of the cameras' intrinsic properties, which are calculated using camera calibration techniques.

2.5 Camera Calibration

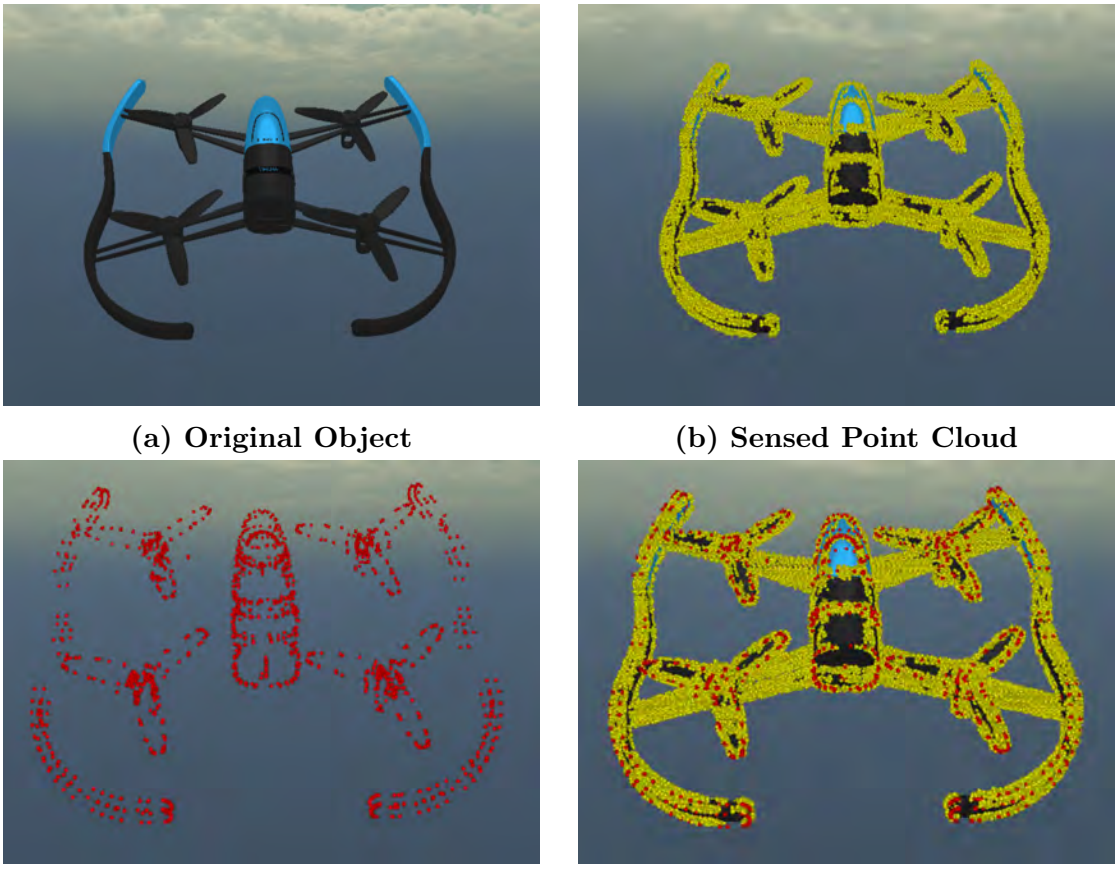
In order to produce the critical points and matrices, the cameras must first be calibrated. The camera calibration calculates both the intrinsic and extrinsic parameters. The intrinsic parameters contains the focal length and principal point of a camera. When calibrating stereo cameras, it is necessary to use image pairs taken simultaneously. The cameras intrinsic information is used to produce the extrinsic parameters. These parameters define the cameras orientation and location relative to the primary camera. When properly calibrated, stereo vision can re-project image points into 3D space using the re-projection matrices.

Camera calibration is performed by taking multiple images of an object with a pattern. Generally, this object contains a checkerboard pattern similar to the one used in [10]. In this application, a black and white checkerboard is used. The square corners are easily detected by vision algorithms. The corners must be a fixed distance from each other. The distance is an important parameter used in the calibration process. The method described by [10] uses the global Levenberg-Marquard Optimization algorithm, minimizing the re-projection error of each corner.

2.6 Registration

Properly calibrated cameras can produce a sensed point cloud that represents object locations. In computer vision, registration is the matching of 2D or 3D reference

point clouds to the sensed point cloud [8]. The reference model of the object must first be created in order to match points. The reference model represents the desired object as a series of points. The reference model points are matched to the sensed points producing an estimated pose of an object, as seen in Figure 4. This registration can be done in two ways, rigid or non-rigid. Non-rigid registration allows for the object vertices to be warped before matching while rigid registration does not.



(a) Original Object **(b) Sensed Point Cloud**

(c) Reference Model Point Cloud **(d) After Successful Registrations**

Figure 4. Registration Process

A fundamental method for rigid 3D registration is the iterative closest point (ICP) algorithm presented in [11]. Every point in the sensed model is matched with the closest point found on the reference model. The reference model is then put into a translation and rotation based on this matching. The process repeats, minimizing

the mean squared error. Once a threshold error or maximum iterations is reached, the ICP algorithm stops.

The ICP algorithm converges to a root mean squared (rms) error of 0. This does not mean that it will find the correct registration or global minimum. It is possible that ICP will settle on a local minimum, generating an incorrect final registration as seen in Figure 5. When stuck in a local minimum the algorithm will not find the global minimum. Methods to break and avoid local minima are provided in [12]. It is also possible to start the reference model in a good orientation to improve the accuracy of the algorithm. The speed of the algorithm was improved in [13] with the implementation of K-D tree data structures. This method tracks points, resulting in faster comparisons for ICP. Further improvements in the speed and accuracy of ICP are found in [14, 15, 16, 17, 18].

2.7 Previous and Related Work

There have been many approaches toward the AAR problem. Differential GPS is one of the methods that has proven beneficial. In [19], a flight test successfully achieved AAR using the probe and drogue refueling method with differential GPS. Likewise, [20] performed a mock aerial refueling using one aircraft acting as the receiver and another acting as a tanker. Both of these tests required the use of GPS and depended on its availability.

Other research has used a combination of differential GPS with other technologies. These technologies include inertial measurement units (IMU), Light Detection and Ranging (LIDAR), and computer vision. It is possible to combine the GPS, IMU, and computer vision data into one estimate using an Extended Kalman Filter (EKF). Combining the GPS and IMU data in [21] with supporting flight test data proved to be beneficial. This approach still greatly depends on the availability of GPS. Computer

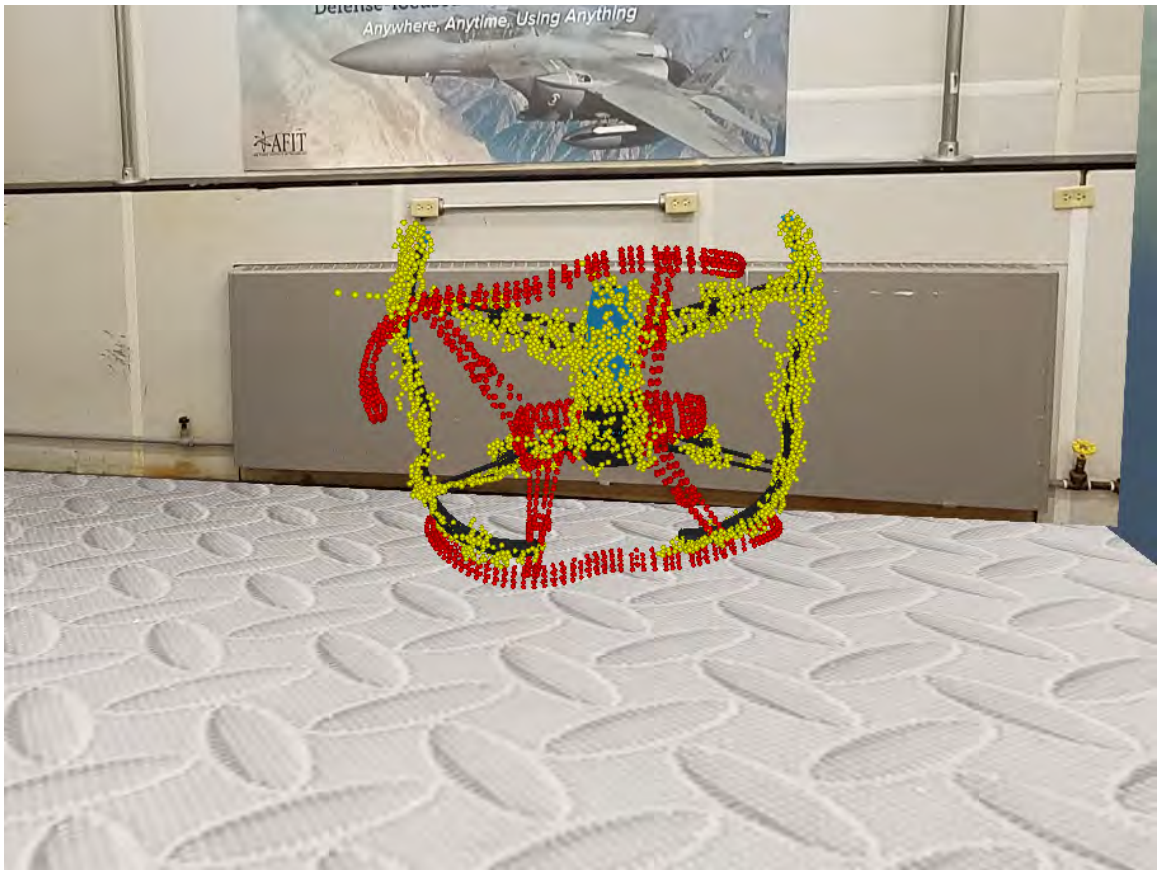


Figure 5. Reference Model in a Local Minimum

vision was fused with GPS in [22, 23, 24] using an EKF. These approaches required the addition of markers to the aircraft. In some cases additional cameras were attached to the receiver aircraft. Additional research in [25] combined GPS, IMU, and computer vision using an EKF. This method of tracking still required additional hardware that is not found on current aircraft in the USAF.

LIDAR was used in [26] to create a point cloud using lasers. This research measured distance from the tanker using a scanning LIDAR mounted on the receiver. This method proved to be useful yet added additional hardware to the receiver. It also poses a security threat to stealth missions if an aircraft uses lasers over what could be hostile territory. Work by [27] used sensors on the receiver paired with lights on the drogue.

Computer vision can use feature detection or point matching to assist in pose estimation. The speed and accuracy of both point matching and feature matching using monocular vision was quantified in [28]. Feature matching with monocular vision was used in [29] and [30] when assessing its value for the AAR problem.

Binocular vision was used in [31] to compute receiver pose. Special markers were added to virtual aircraft allowing speeded up robust features (SURF) to be extracted. The markers, paired with SURF, help process the receiver pose. Building off of this virtual experiment, a real world experiment conducted in [32] used binocular vision paired with saliency maps to help calculate the pose of a receiver. The saliency maps produce easily identifiable features. The aircraft used in this experiment were two small UAV copters imitating a tanker and receiver. Both the virtual and the real world experiments required adding markers to the receiver.

Work on the AAR problem was performed at AFIT by [33, 34, 35]. These approaches use stereo vision solutions for the AAR problem. Denby uses a Vicon chamber as the environment for the mock refueling approach. A Vicon chamber measured the distance between a stereo rig approaching a scaled down aircraft. The imagery was processed in real time, but did not provide an estimation accurate enough for AAR. Work done by [35] used a virtual world as well as virtual cameras in order to mimic the real world. A refueling flight approach was captured using synthetic imagery. The tanker boom was removed from the model to produce an unobstructed view of the receiver. The reference model used was shelled in a way that the bottom half of the aircraft was not included. This produced real time pose estimation of the aircraft with less than 20cm error.

The AAR problem has been tested in both the real world and in virtual environments. The research done in [25, 28, 29, 36] used MATLAB Simulink and Virtual Reality Toolbox. These environments do not produce imagery that closely resembles

the real world. Work in [31] used real world cameras with virtual imagery by directing the cameras at two different monitors. Werner used the Ogre 3D graphics engine in [34]. While the virtual imagery proved beneficial for pose estimation, the methodology did not imitate the real world flight paths or the same hardware already existing on the KC-46. Virtual environments proposed in [37, 38] do not provide the capabilities to use computer vision. Research done in [39, 40] use a Kinect sensor to map out real world objects into a virtual world. This is all done using KinectFusion. This technology use simultaneous localization and mapping (SLAM) to produce a virtual scene in real time. The 6DOF of the Kinect sensor must be constantly tracked in order to create an accurate reconstruction of the real world. The sensor also uses an infrared depth sensor that does not work in outdoor environment.

Vision sensors are used in several different aspects for pedestrian and vehicle detection using neural networks. Training neural networks takes large data sets of imagery. Work in [41] uses a deep learning for vehicle detection. The architecture is trained exclusively on synthetic imagery. Once trained, real world imagery of vehicles is tested. A scene specific pedestrian detector is used in [42]. The scene is modeled after the location that is under surveillance. Modeled pedestrians are then placed in the scene. The camera parameters are used to create image distortion on the pedestrians as if it was the real vision sensor being used. The trained network is then tested on the real scene. Similarly, [43] evaluates the possibility of training pedestrian detection using only synthetic imagery. Two neural networks are trained exclusively on real or synthetic imagery. These trained neural networks are then tested on the same set of imagery. A high correlation is found in performance of both trainings as well as a correspondence in detection results. Stop sign distance is measured in [44] using a virtual world. Single synthetic images are taken from the virtual world at several ranges. The Grand Theft Auto V engine is used in [41, 44]. This engine

can accurately model the real world with photo realistic imagery as well as several weather conditions, lighting effects, and hundreds of human and car models.

Several experiments have compared the use of synthetic imagery to replace real world imagery. These experiments were mainly used to train neural networks and not for pose estimation. The work performed toward pose estimation has all been either in a simulation or real world experiments. It is apparent the use of real world and synthetic imagery for pose estimation needs to be quantified. Real world vision experiments recreated in a virtual world would greatly benefit the AAR research. This would provide a comparison in performance in both environments.

III. Methodology

Comparing computer vision in the real world and the virtual world require several things to be closely related. These include the object, the environment, the real and virtual camera parameters, the calibration output of those cameras, the registration models, and the computer vision pipeline. The object and environment are easily imitated using modeling software. The real world camera parameters can be implemented into the virtual sensors. These can both be calibrated using checkerboards. The calibration outputs can be compared to see how closely they relate. Computer vision is managed by Open Source Computer Vision (OpenCV) libraries [1].

The drown follows several different flight paths that cover the majority of the viewing volume. The motion is precisely tracked via the Vicon chamber. The drone and Vicon environment are accurately modeled in the virtual world. The drone follows the logs from the Vicon chamber to recreate the real world experiments. Both the real and virtual experiments are captured using stereo vision. The estimation of the drone location will follow the same computer vision pipeline for both real and synthetic imagery.

3.1 Environments

The real world can be replicated in the virtual world. Objects can be modeled and rooms details can be textured. In order to accurately compare the performance of ICP on real world and virtual world data, the environments must match.

Real World.

The real world experiments are conducted in a Vicon chamber. This chamber uses eight infrared cameras to track spherical markers attached to objects. The multiple

cameras enable the system to accurately track an object's position and orientation. In static and dynamic studies, [45] was able to achieve mean errors of 0.15mm for static objects and less than 2mm error for dynamic objects depending on their speed. After Vicon calibration, a similar mean error less than 2mm was calculated for this research.

Collecting data for experiments involves several systems. A collection machine is used to collect and process images from the cameras. The raw Vicon data is collected through a different windows machine. This machine is controlled by a linux laptop via ethernet. The laptop receives raw Vicon data and turns it into human readable numbers. The laptop is also running an NTP server that syncs the collection machine and Vicon machine clocks. Timing is important to correlate images with the correct Vicon position information.

The object being tracked is a Parrot Bebop quadcopter seen in Figure 6. The quadcopter was attached to a wooden pole via fishing line. This configuration not only helped the drone follow specific paths, but allowed the drone to consistently "fly" while rotated 90 degrees. This orientation forced the top of the drone to face the cameras, revealing more object points to maximize the sensed point cloud. A larger sensed point cloud provides better matching for ICP. This view is similar to a top down view of the drone which can be comparable to a tanker looking down at a receiver. The propellers of the drone were removed providing a consistent model. Moving propellers change the objects shape, thus requiring a dynamic reference model. It is also assumed that in a real flight the propellers will be moving too fast to be captured by the cameras. This phenomenon was observed during recent flight tests conducted in September of 2017.

The drone has several infrared markers attached to it. This allows the Vicon chamber to track its movements at 100HZ. The drone position and orientation is

Figure 6. Parrot Bebop Drone

logged with the time stamp, XYZ position, and a quaternion. Unlike previous research, these markers are not used to aid in the estimation of the pose of the aircraft. This data is used as the truth data as well as the log file to recreate the flight in the virtual world.

Virtual World.

The virtual world is the same OpenGL based AFTR Burner engine used in [35]. This engine uses high fidelity models and textures that represent the real world [46]. The AFTR Burner engine is also capable of visualizing the output of synthetic sensors [47]. These sensors can be specified to match the parameters of real world cameras. As the receiver pose data is updated, the sensors render a stereo pair of images.

The Vicon chamber was modeled using the floor and back two walls. The cameras

face these walls when collecting data. The real world Vicon chamber and re-creation can be seen in Figure 7. The virtual drone is a scaled model of the Parrot Bebop drone. Due to the small size, the Vicon tracking spheres were not modeled on the virtual reference model.

(a) Vicon Chamber **(b) Virtual Vicon Chamber**

Figure 7. Recreating the Vicon Chamber

3.2 Stereo Vision

Stereo vision requires the use of two cameras in order to create a point cloud. In this research, the cameras are set parallel to each other. Due to limitations on space, the camera's baseline could not match the .5 meter baseline used in other AFIT research [35, 48, 49]. The baseline chosen fell between the average pupil distance of men and women found in [50]. This baseline is small enough to scale down the distance of the refueling position, without being to close to estimate depth accurately. After calibration, the distance of the virtual and real world cameras was near 63.5mm. The cameras used were Prosilica 1290 electro-optical cameras seen in Figure 8. The position and orientation of the primary camera was captured using the Vicon spheres. The aspect ratio of these cameras is 1280x960. They are triggered using a BNC cable. The images are transferred using an ethernet cable directly connected to the collection

machine. The transfer is done using User Datagram Protocol (UDP). While UDP usually doesn't resend packets, there is an on camera buffer that stores images. In the event that a packet is lost, the collection machine queries the camera and the camera will resend. The captured data is 24 bit, RGB images. The cameras are triggered at a rate of 30hz. For this rapid data transfer, each camera had a dedicated gigabit line to the collection machine. The collection machine can process this imagery using 24 cores before saving it to the 4 terabyte solid state drive.

Figure 8. Stereo Cameras with Vicon Spheres

The use of stereo vision to track a moving object requires the left and right cameras to take pictures simultaneously. This requirement is easily programmable in the virtual world, but necessitates extra hardware for the real world cameras. Syncing the cameras in the real world can be achieved using the triggering board seen in Figure 9. This board receives a signal from the collection machine using a 3.3 volt pulse from a usb cable. This cable sends a 30hz signal to a tri-state chip on the board. This

signal branches to the two cameras using a BNC cable from this board. The board and triggering was tested for drifting between the cameras. This test consisted of the two cameras pointing at a 60hz monitor. The monitor displayed a stopwatch with millisecond precision. The cameras captured images at 30hz for tests of 5 minutes, 10 minutes, and 60 minutes. At the end of each test, several image pairs were picked out, prioritizing images near the end. The displayed times all matched on the image pairs, confirming no drift or delay in the triggering or processing of the imagery.

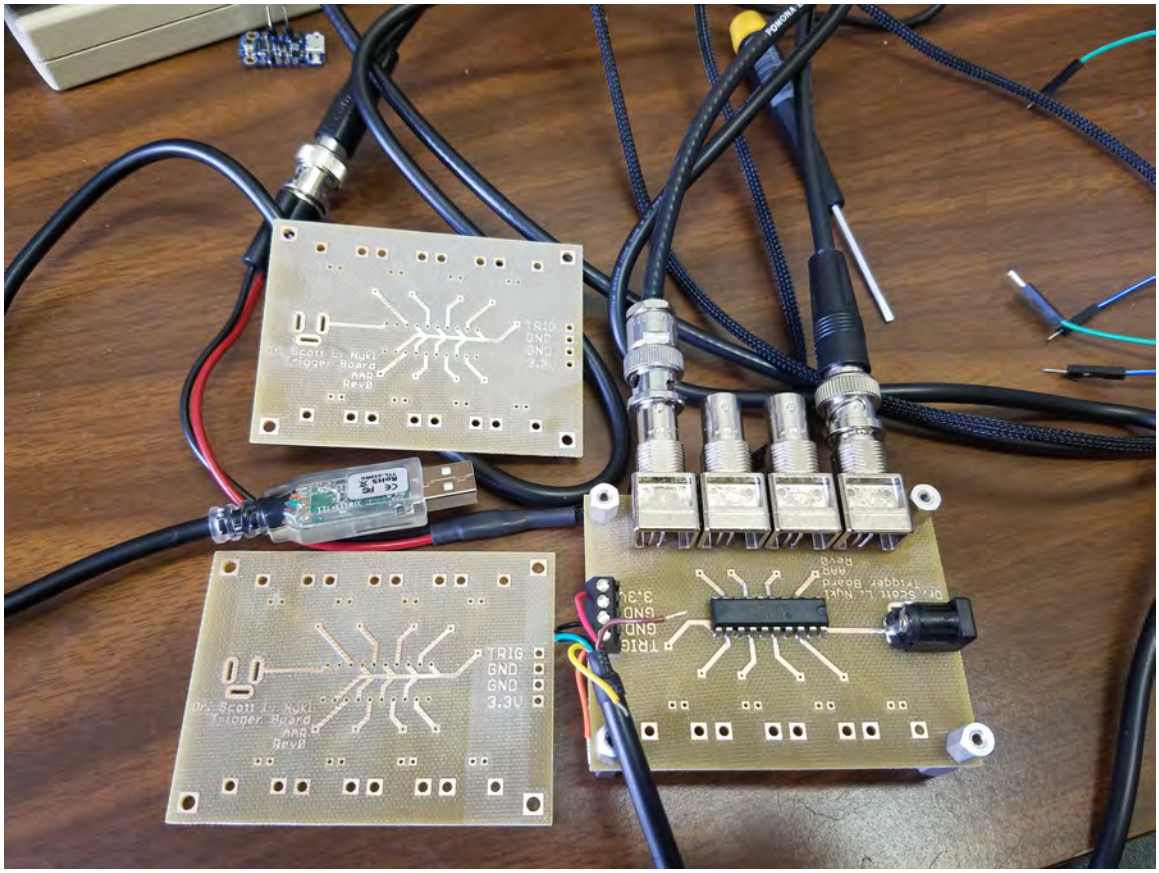
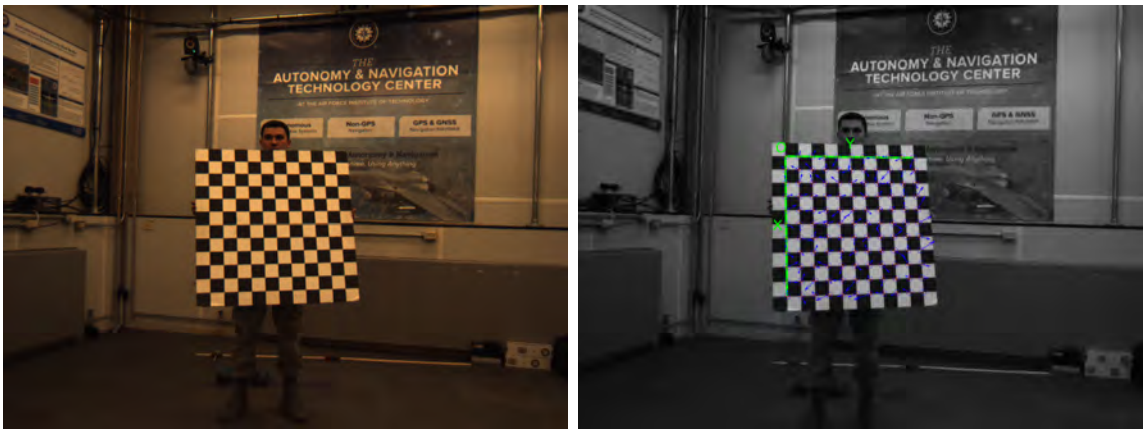


Figure 9. Triggering Bord for Syncing Cameras

3.3 Calibration

The calibration process used a checkerboard with known distances between the squares. The checkerboard was put in different locations and orientations to cover the

viewing field of the cameras. The cameras simultaneously captured these positions. The real world images were processed through the Camera Calibration Toolbox for Matlab [51]. This calibration process requires the user to pick the outer four corners of the checkerboard. Once the boundary is set, corner detection is used to find the remaining corners. Once all the images were processed, the intrinsic parameters were calculated. Figure 10 shows the original picture as well as the picture after corner detection. When both the right and left cameras' intrinsic parameters were found, the toolbox calculated the extrinsic parameters. A display of the stereo extrinsic parameters, including projections of the checkerboard locations, can be seen in Figure 11. The cameras were calibrated everytime the stereo camera rig was relocated. Over 20 image pairs were used in every calibration to produce sub pixel error.



(a) Original Image **(b) Corners Detected**

Figure 10. Corner Detection Using Matlab

The virtual world used a similar process for calibration. A black and white virtual checkerboard was moved around the viewing area of the virtual sensors. Bitmaps of these positions were captured. Using OpenCV, corner detection was performed and the intrinsic and extrinsic parameters were calculated. The checkerboard image along with corner detection of that image can be seen in Figure 12. A total of ten image pairs were used for calibrating the virtual cameras to produce sub pixel error.

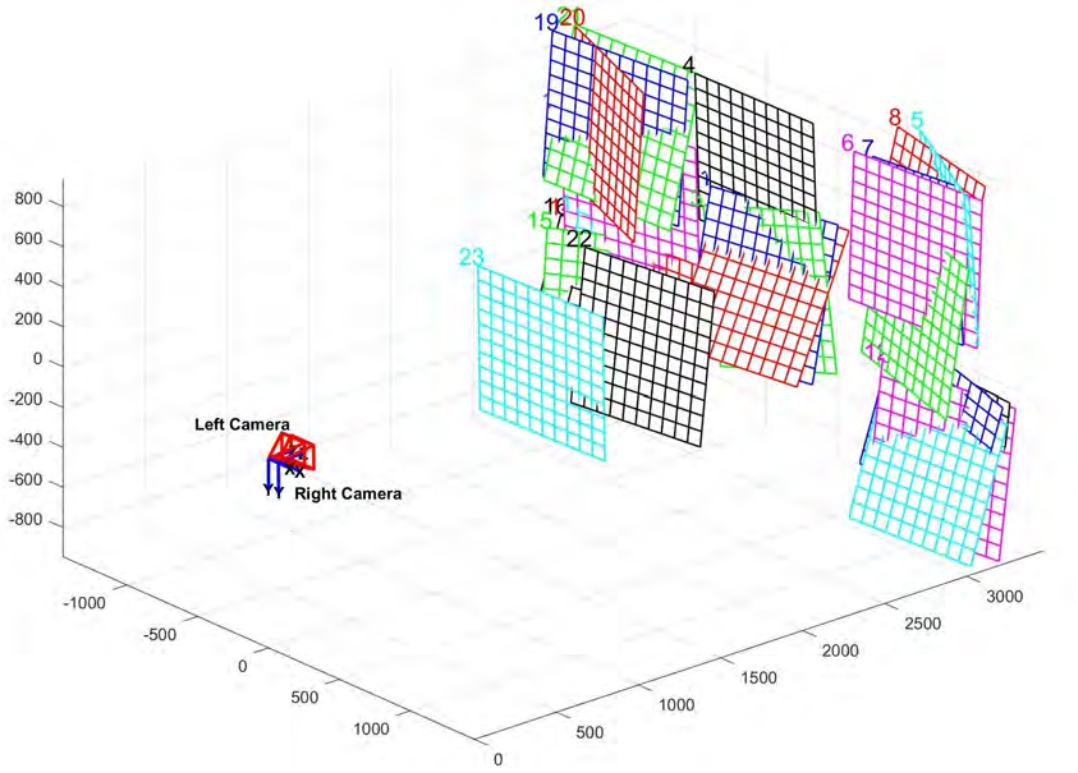
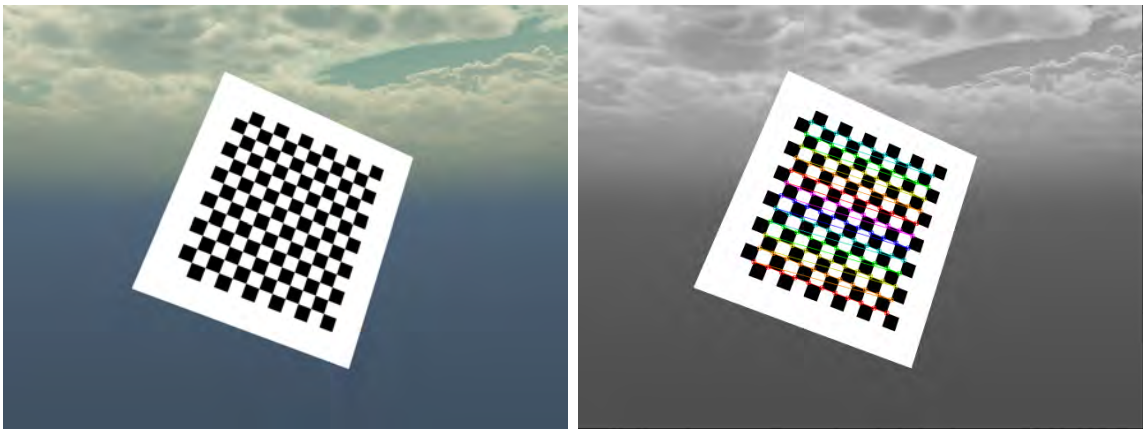


Figure 11. View of Extrinsic Parameters



(a) Virtual Checkerboard **(b) Corner Detection**

Figure 12. Corner Detection on Synthetic Imagery

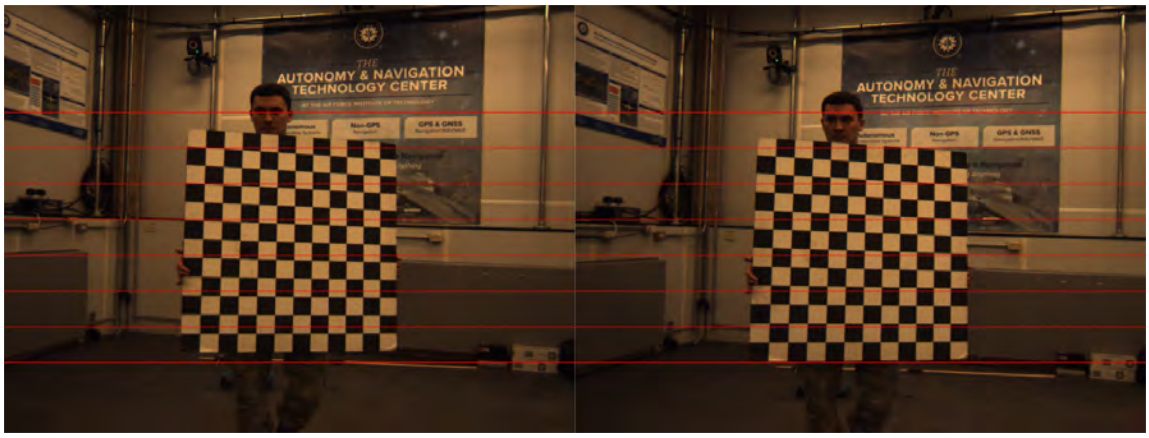
Stereo calibration produces parameters that create a Q matrix. This matrix is necessary for the computer vision pipeline. Additional matrices proposed in [10] are also produced in both camera calibration methods. The epipolar constraint error is

computed using the essential matrix. The average epipolar error is calculated using the essential matrix with the left and right detected image points, $p1$ and $p2$.

3.4 Disparity Map

With proper calibration parameters, the stereo vision pipeline produces a disparity map. This is done using OpenCV's block matching function. The block matching function is simplified when using rectified images. Image rectification aligns corresponding epipoles to be on the same horizontal row in the left and right images. A point, P, on an object can be represented by P_r and P_l, in the right and left image. The pixel distance between these image points is the disparity. Rectification skews the left and right images in such a way that the pixels are on the same horizontal row, as seen in Figure 13. This process reduces the feature matching search space to a single row, speeding up disparity map generation. The disparity map is relative to the left camera, remaining consistent with the re-projection matrix Q.

OpenCV provides adjustment to the total number of disparities as well as the window size for block matching. These parameters have an effect on the reliability and accuracy of the point cloud. The number of disparities limit the maximum search range in pixels for feature matching [1]. The window size defines the size of the pixel block being matched. A larger window size reduces the number of details that are included in the point cloud, resulting in less sensed points. A smaller window size will include finer details, however opens up the potential for more mismatches. It was found in [35] that a window size of 9 and a number of disparities equal to 48 produce a balanced approach. Disparity maps can be seen in Figure 14. The lighter the pixel, the closer the point is to the camera, where black sections are the farthest away. Figure 14a shows the disparity map for the virtual drone up close with no background. Figure 14b is a disparity map of the real world. The Vicon chamber can



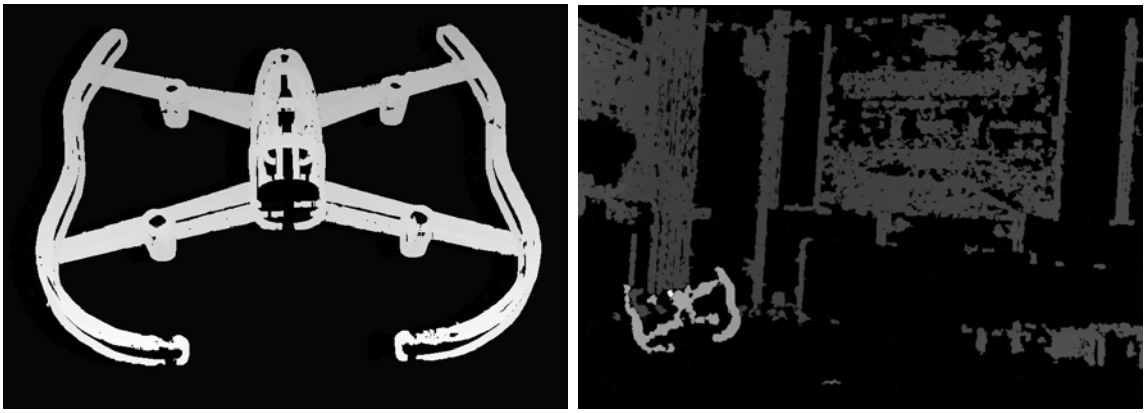
(a) Original Image Pair

(b) Rectified Image Pair

Figure 13. Image Rectification

be seen in background while the drone is lighter object in the foreground.

It is possible for outliers to exist in the disparity map. Outliers are a result of mismatched object points and have a negative effect on registration. To combat this, a speckle filter was applied. This filter removes these outliers if they do not meet a set threshold. It is possible that the filter may remove points off of the drone object, however there still exist enough points to perform ICP.



(a) Just Drone **(b) Drone with Vicon Background**

Figure 14. Disparity Maps

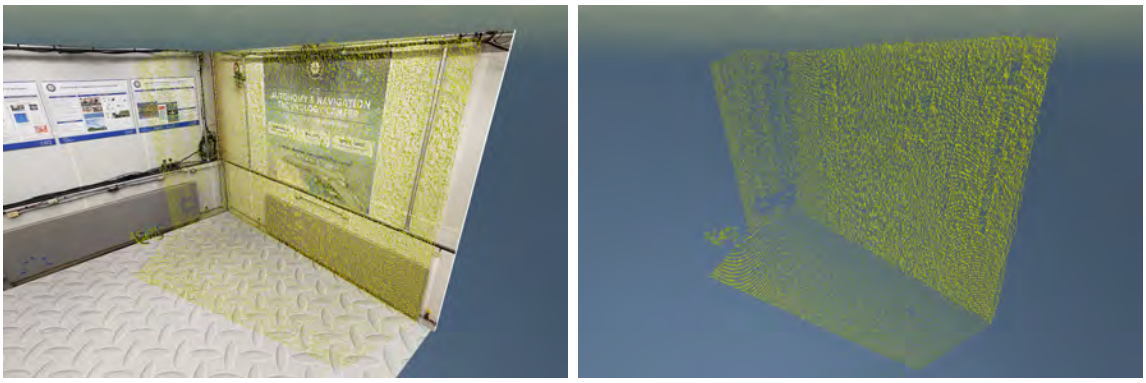
3.5 Re-projection and Point Cloud Generation

Once the disparity map is generated, the relationship between the left and right imagery can be visualized. The disparity map coupled with the transformation Q matrix enables the points to be reprojected in 3D space. To visualize points produced by real world imagery, a re-projection was created in the virtual world. OpenCV computes the x, y, and z components for each pixel point (p_x, p_y) [1] using Equation 1:

$$[xyzw]^\top = Q * [p_x \ p_y \ disparity(p_x, p_y) \ 1]^\top \tag{1}$$

This equation produces a matrix of 3D points that represent the sensed point cloud. This matrix includes all sensed points, including points that are not the desired object. Figure 15 shows a full point cloud that is generated. The first picture includes the environment that is being projected, while the second picture is just the point cloud. It can be seen that the walls are included in this point cloud. In order for ICP to work properly, it is necessary that only the drone point cloud is projected.

This was accomplished using the truth data that was provided by the Vicon chamber. Points within half a meter of the drone location were the only points projected. It is assumed that during an aerial refuel, any other objects will be far behind the aircraft. The far points can be filtered, leaving the aircraft as the only point cloud visible. Previous research filtered the point cloud at a specific cut off distance.



(a) Point cloud with the objects **(b) Point Cloud**

Figure 15. The Full Point Cloud Before Filtering

An additional filter was applied reducing the number of projected points to one fourth of the total. This reduction increased the speed of ICP allowing for real time registration. It was observed in [35] that reduction of the total point cloud size still retained registration accuracy. Once the point cloud is reduced, it must be transformed to the Vicon coordinate frame. The point transformation from the sensed frame to the Vicon frame $[x_v\ y_v\ z_v]$ can be seen in Equation 2.

$$[x_v\ y_v\ z_v] = [z\ (-x)\ (-y)] \tag{2}$$

After this transformation, the points must be transformed into the position and orientation of the left camera. The cameras position and orientation is applied to each point in the point cloud. The transformation is done using the left camera direction cosine matrix DCM_{camera} and the left camera position $[x_c\ y_c\ z_c]$. The new points

$[x'\ y'\ z']$ are produced by Equation 3.

$$[x'\ y'\ z']^\top = \left[(DCM_{camera})^\top * [x_v\ y_v\ z_v]^\top\right] + [x_c\ y_c\ z_c]^\top \tag{3}$$

Once the point cloud was filtered and transformed, it was used for ICP. The filtered point cloud produced by synthetic imagery can be seen in Figure 16. The view is from the perspective of the left camera.

(a) Point Cloud in Vicon **(b) Drone Point Cloud**

Figure 16. Point Cloud After Filtering

3.6 Model Registration

The ICP algorithm is used for registration of the sensed point cloud with the reference model. A point-to-point ICP implementation with a modified KD-Tree is used to converge the reference point cloud with the sensed point cloud [18]. The rotations and translations at each iteration are tracked until the rms threshold is reached or 30 iterations are completed. A successful ICP registration can be seen in Figure 17. The yellow dots represent the sensed point cloud, while the red dots represent the reference model. The reference model was created using a reduced set of vertices from the virtual drone model.

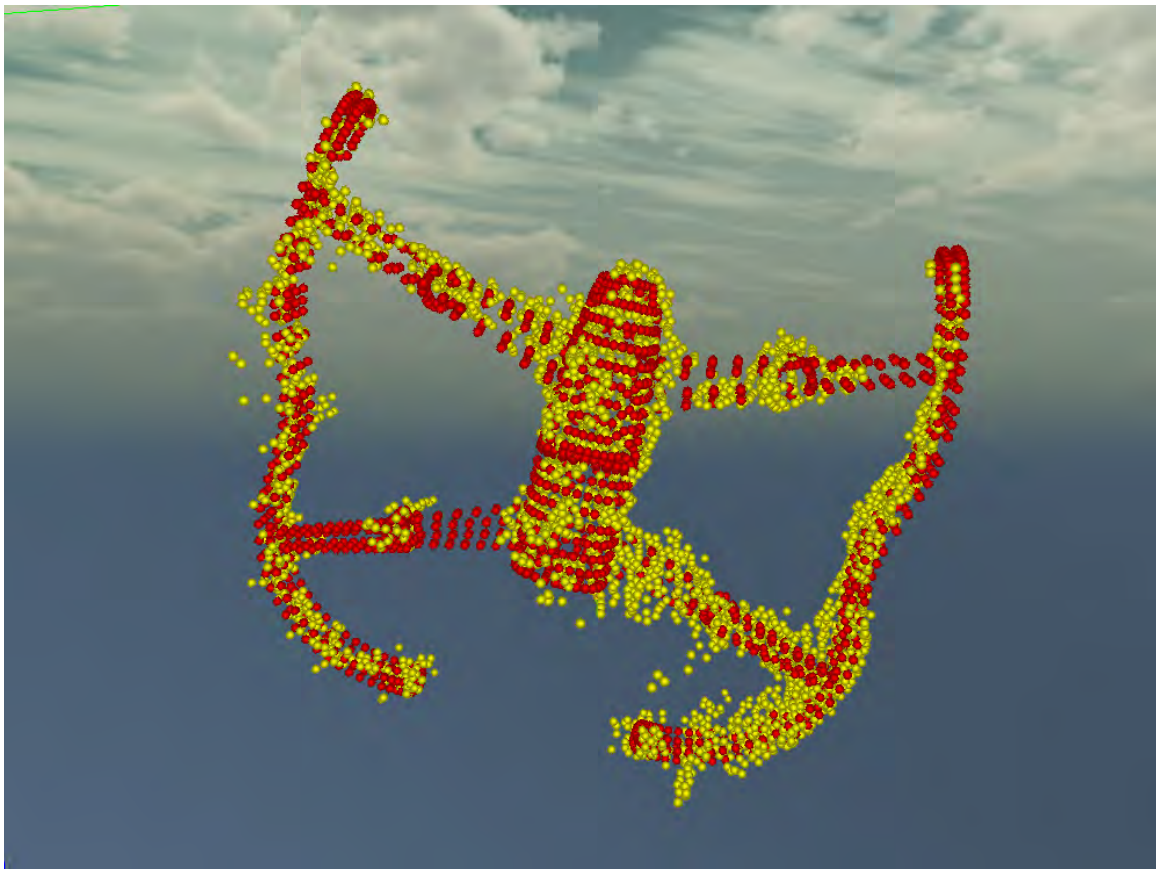


Figure 17. Successful ICP Registration

The shape of the qudcopter drone generates the potential for several local minima. For this reason, the estimated orientation after ICP did not always correlate with the truth data. This was observed regardless of real or synthetic imagery. In this research we assume that the drone's orientation will provide a top down view. This assumption is based on the fact that a quad copter cannot sustain flight in another orientation for an extended amount of time. For this reason, the primary focus of this research is on the position of the reference model after ICP.

3.7 Experimental Design

This experiment measures the accuracy of the ICP algorithm as performed with real and synthetic imagery. The estimated positions were computed and compared

to the truth data provided by the Vicon chamber. Statistical analysis was performed on the errors produced in both environments.

Experiment.

Using a Vicon chamber, a quadcopter drone was suspended using fishing line. Several tracking markers were attached to the drone. The drone then performed six pseudo-flights in the stereo camera frames. The cameras captured the drone's flights at 30hz. When processed, a disparity map of these images was generated. With this information, a sensed point cloud was calculated and filtered. ICP registration was performed on the filtered point cloud and the position of the reference model estimation was recorded. The estimated position was transformed into the coordinates relative to the primary camera. Once transformed, the estimated and truth positions were used to find the error. These errors were used to compare the use of real and synthetic imagery.

The computer vision pipeline is very similar in the virtual world. The difference exists when updating the drone position and capturing the images. The virtual drones position followed the logs produced by the Vicon chamber at 100hz. This log contains the exact position and orientation of the drone relative to the chamber origin. The drone was a one to one scaled model of the real world drone. This re-creation of the real world experiment was captured by virtual stereo cameras. These images then follow the same computer vision pipeline as the real world imagery.

The computer vision pipeline can be seen in Figure 18. After the camera calibration and stereo pairs were captured, the pipeline is identical. The real world imagery was replayed in real time while the synthetic imagery was captured and immediately processed. Captured image pairs generate the disparity map and a filtered point cloud is projected. This point cloud is visualized in the virtual world regardless of image

source. ICP registrations were performed with a maximum of 30 iterations. The 30 iterations allow for a time bounded stopping point. If the RMS error is lower than the previous stopping point, ICP may terminate before 30 iterations.

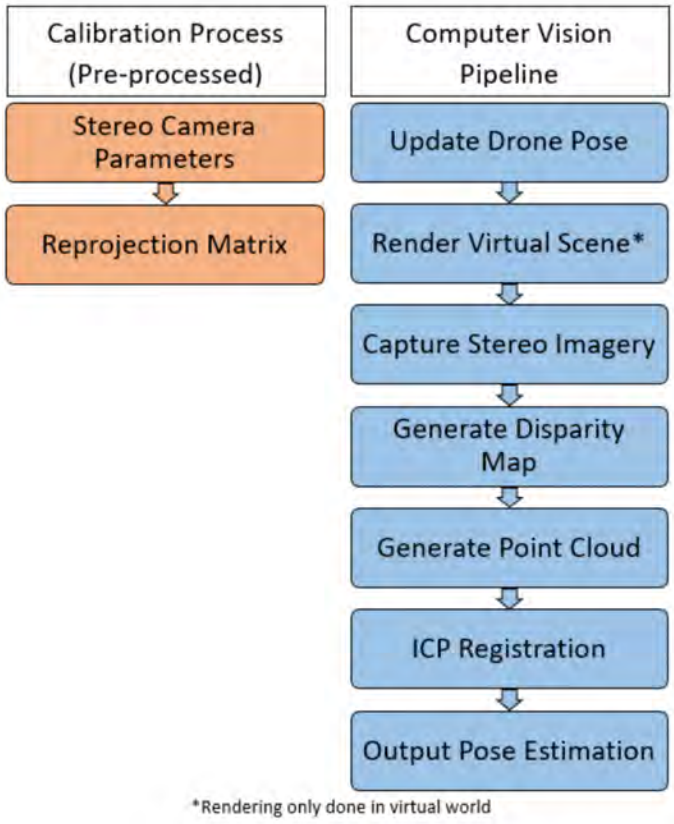


Figure 18. Computer Vision Pipeline

Error in the estimated position was used for statistical analysis. Statistical analysis included the mean, standard error, median, standard deviation, variance, range, min, and maximum. Additionally, the root means squared deviation (RMSD) was also calculated to show similar trends in behavior. The RMSD is calculated using Equation 4, where n is the number of samples and $\hat{v}_p$, v_p are the estimated and truth positions respectively.

$$RMS\ Error = \sqrt{\frac{\sum_{n}^{p=1}(\hat{v}_p - v_p)^2}{n}} \tag{4}$$

IV. Results

The AAR problem relies on the 6DoF estimation of the receiver when in the refueling envelope. This envelope represents a 3D space where a receiver can obtain fuel. It is possible for the receiver to approach from behind the tanker or from the side of the tanker. For this reason, an omni-directional range needs to be evaluated. This research directly relates to the AAR problem by testing a three dimensional region. The region is restricted by the camera's 56 degree field of view, as well as the walls of the Vicon chamber. The drone covered a range in all directions: 2.07m to 4.33m in the x direction, -1.5m to 0.833m in the y direction, and -0.77m to 0.83m in the z directions. This area can be visualized by the different flight paths in Figures 19 and 20. Each color ribbon represents one of the six flight paths. The rgb lines that create a boundary represent the cameras' frusta.

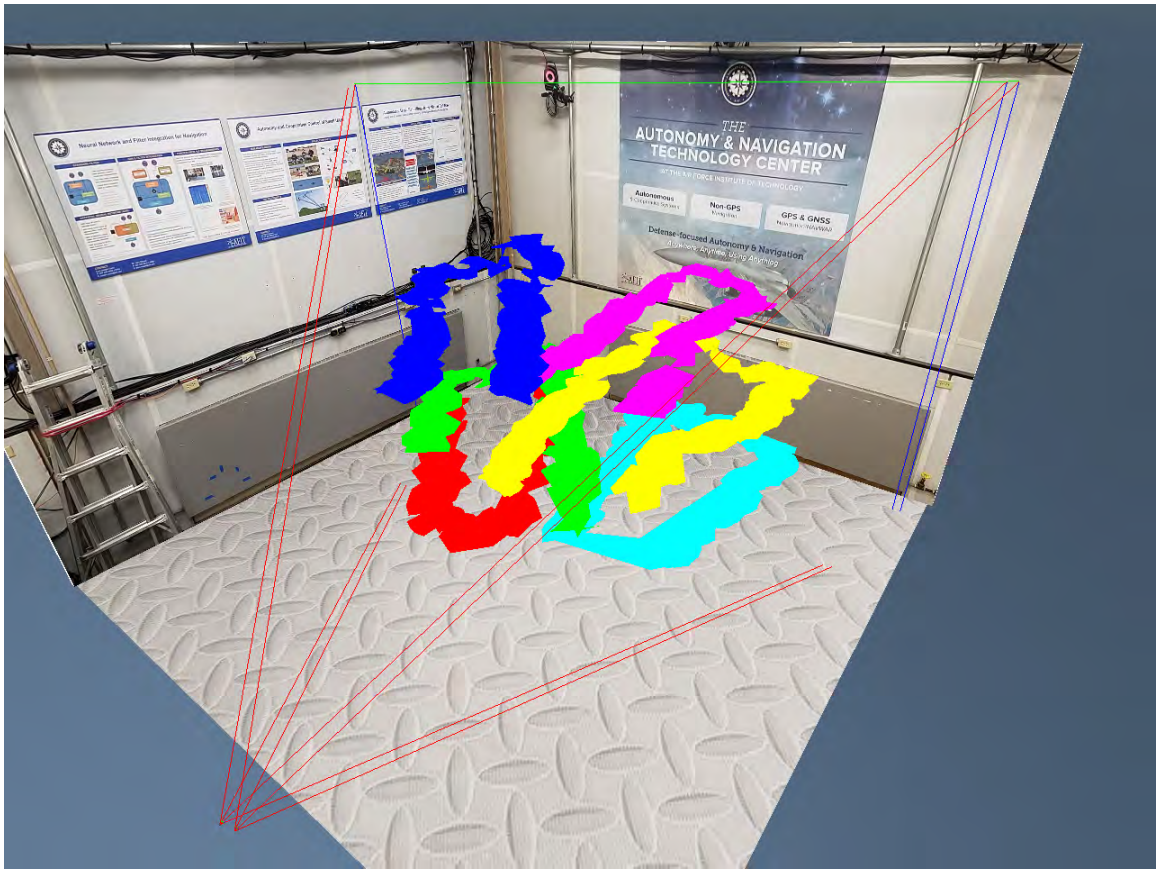
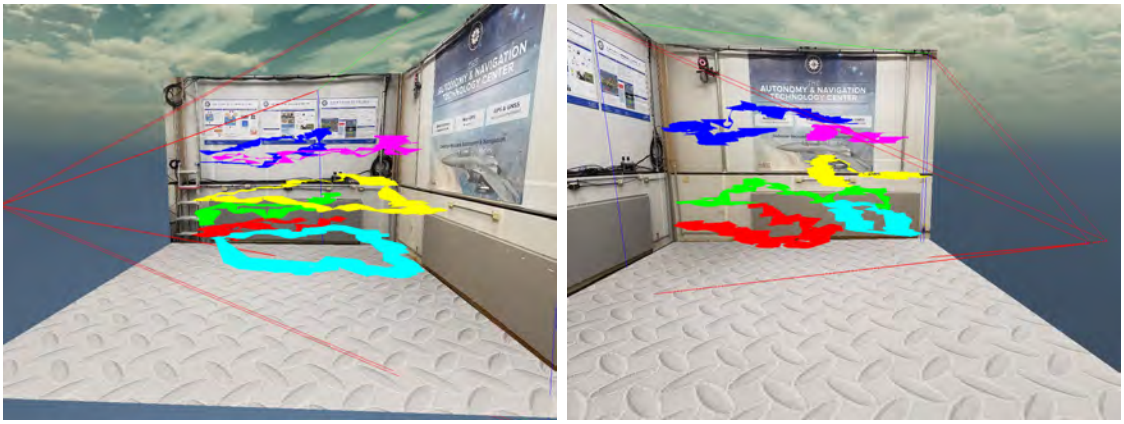


Figure 19. Top View of the Flight Paths



(a) Side View **(b) Side View**

Figure 20. View of Flight Paths in Mock Envelope

4.1 Reference Model Performance

The data analysis on estimations using real and synthetic imagery can be seen in Tables 1 and 2 respectively. These numbers represent the XYZ error for the combined 6 tests performed. Over 2,000 data points were produced for each environment. The mean errors in the X, Y, and Z directions are the main focus. Comparing the tables, the means are all less than 1.5 cm of each other.

Table 1. Data Analysis for Real World Imagery

	X Error	*Y Error*	*Z Error*
Mean	0.0438582	0.02056916	0.0216967
Std Error	0.00045979	0.00032269	0.00033994
Median	0.0451183	0.0175233	0.0182927
Std Deviation	0.02248279	0.0157789	0.0166222
Variance	0.00050548	0.00024897	0.0002763
Range	0.10699402	0.09146509	0.09541177
Minimum	0.0000879	0.00000381	0.0000119
Maximum	0.107082	0.0914689	0.0954237

The potential XYZ estimation error of the vision sensor can be anywhere in the mock envelope. Taking the mean error and dividing it by the potential error gives a percentage that can compare the real and synthetic results. The potential locations based on the envelope can be in a range of [226cm, 298cm, 161cm] in the X, Y, and Z directions, respectively. These comparisons can be seen in Table 3. The estimation error from real world and synthetic imagery are less than 0.6% of each other.

Trends can be seen in the data when it is separated by distance magnitude from the camera. The farthest distance being 4.49 meters and the closest distance being 2.09 meters. This divides the near and far fields into distances less than or greater than 3.29 meters. Taking the RMSD of the observed values, the accuracy of ICP can

Table 2. Data Analysis for Synthetic Imagery

	X Error	*Y Error*	*Z Error*
Mean	0.03074492	0.00827584	0.01361484
Std Error	0.00028189	0.00017192	0.00030841
Median	0.0287218	0.00504518	0.00648254
Std Deviation	0.01446434	0.00882148	0.01582529
Variance	0.00020922	7.7819E-05	0.00025044
Range	0.14940237	0.05903988	0.09571146
Minimum	0.00011063	0.00000222	0.00000134
Maximum	0.149513	0.0590421	0.0957128

Table 3. Percent Error Comparison

	X	Y	Z
Real World Error	1.94%	0.69%	1.35%
Synthetic Error	1.36%	0.28%	0.85%
Difference	0.58%	0.41%	0.50%

be compared in the near and far fields. Table 4 shows that in both the real and the virtual world, the ICP algorithm performs better in the near field. This comparison confirms that behaviors present when using real world imagery are also present when using synthetic imagery.

Table 4. RMSD analysis of the near and far fields

		X	Y	Z
Real Imagery	< 3.29 m	0.02374769	0.02431408	0.06039172
	>3.29 m	0.02710835	0.028942	0.06296416
	Difference (%)	12.4	15.99	4.09
Synthetic Imagery	< 3.29 m	0.02866318	0.00836307	0.01297635
	>3.29 m	0.03717344	0.01408743	0.02486816
	Difference (%)	22.89	40.63	47.82

4.2 Visualizing Error Trends

The X, Y, and Z error can be compared using overlaid histograms. The data is based on the errors produced by ICP in the six flight tests. The total number of estimations for both sources ranged from 2300 to 2600 iterations, with synthetic imagery producing more iterations. Figures 21, 22, and 23 show the error range of estimations as well as the count of the number of errors in that range. It is noteworthy that the X error average is greater than the Y and Z errors. This is expected based on the numbers that are seen in the previous sections. This will be further shown and investigated. The Y and Z errors for both the real and synthetic imagery have a visible relationship. While the synthetic imagery does have a lot more estimations in the lower range, they still have similar distributions.

The truth position and the estimated position both produce individual colored ribbons, as seen in Figure 24. The ribbon with black represents the truth data. The colors correlate to time such that the matching colors are the truth and estimate at

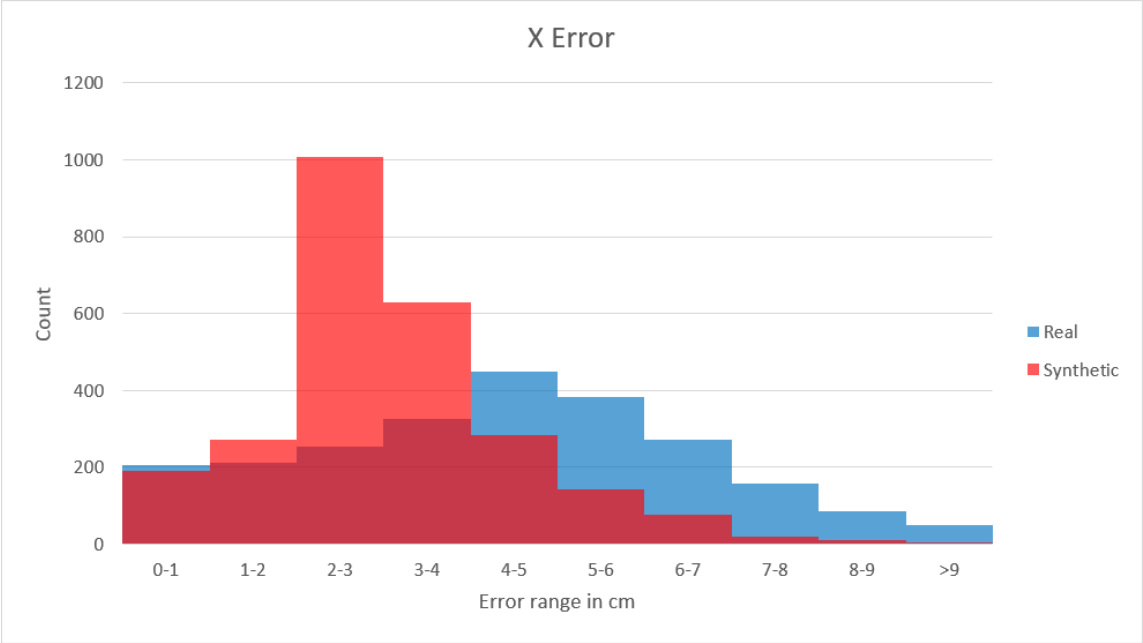


Figure 21. Histogram of X Error

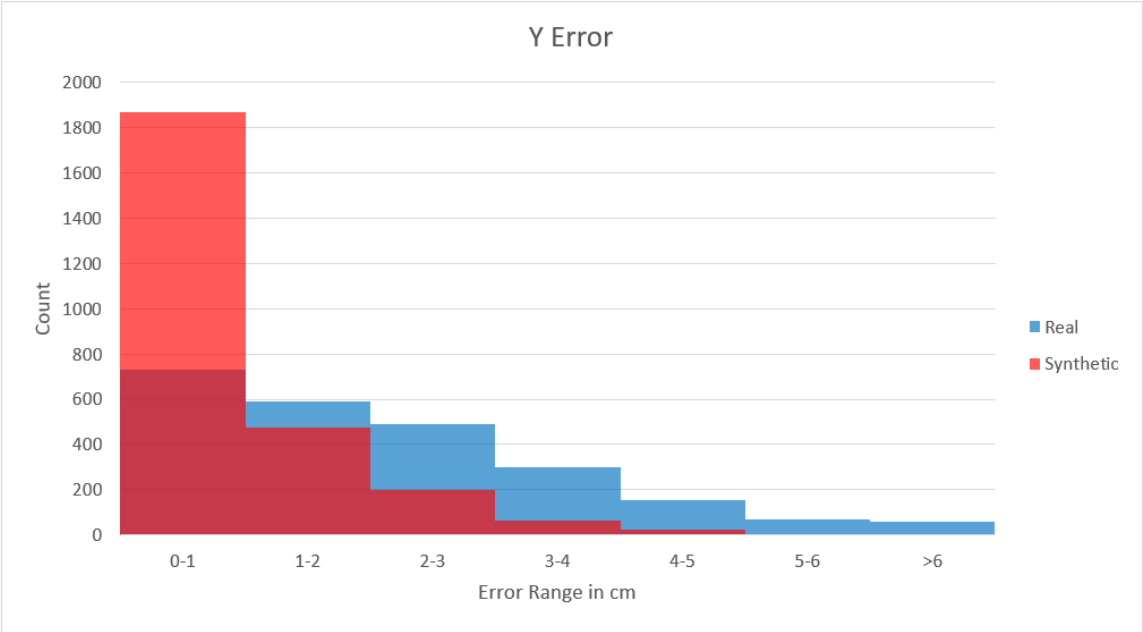


Figure 22. Histogram of Y Error

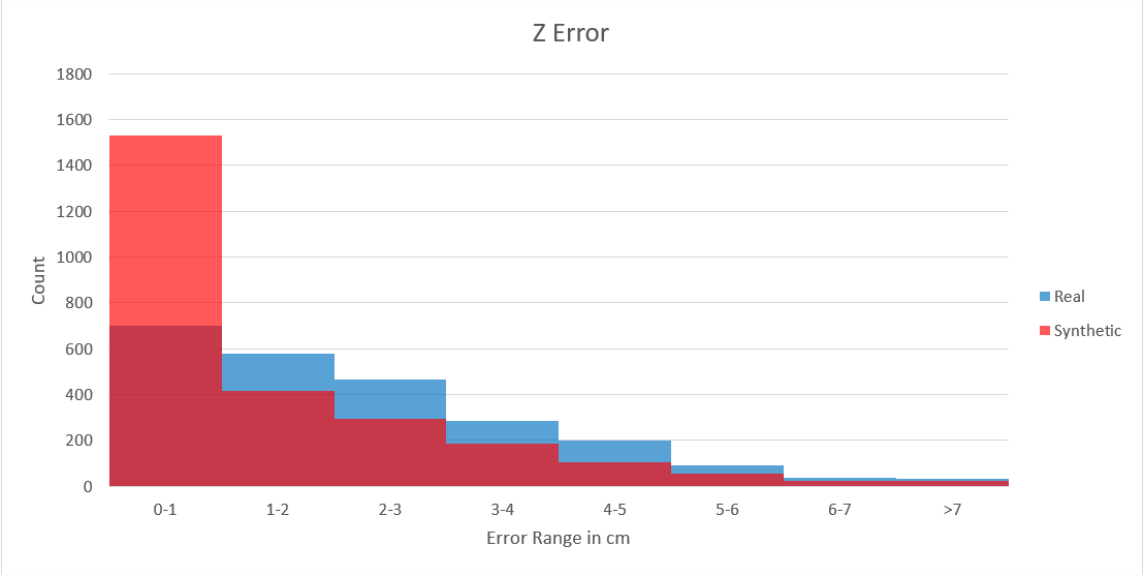
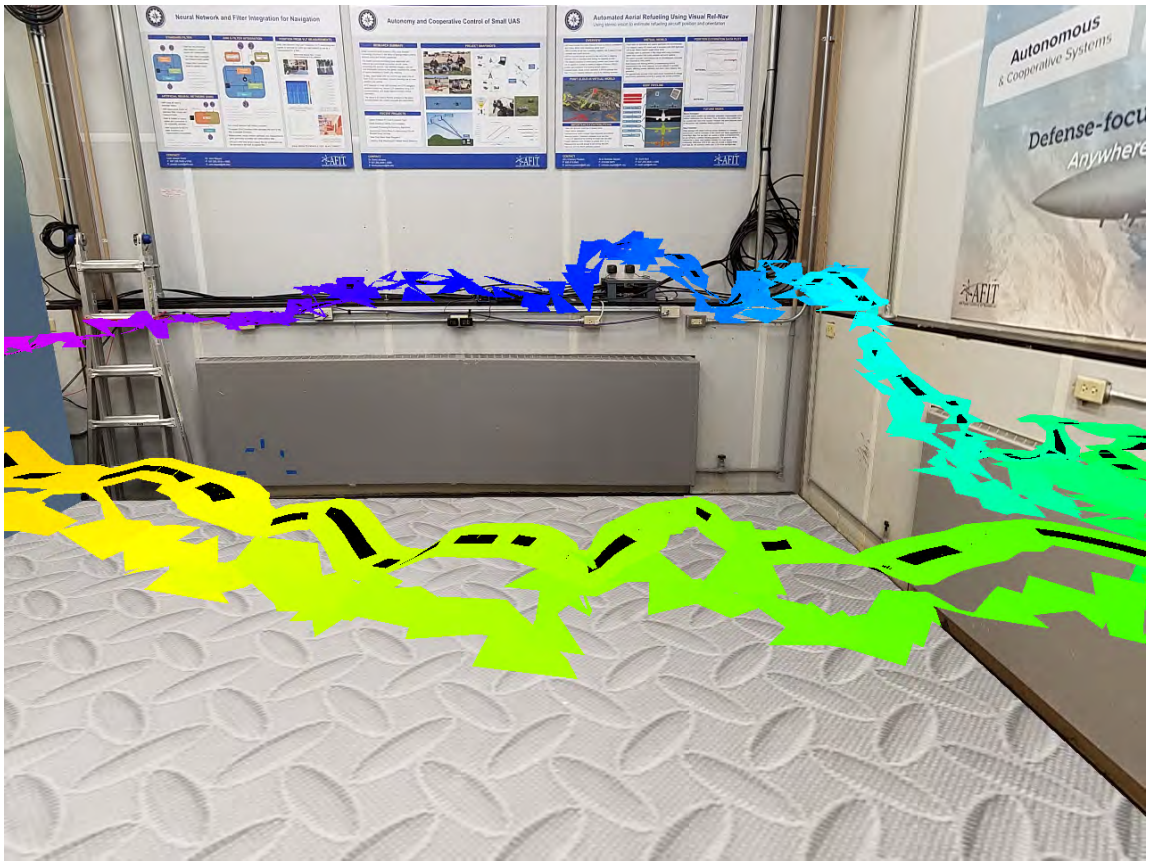
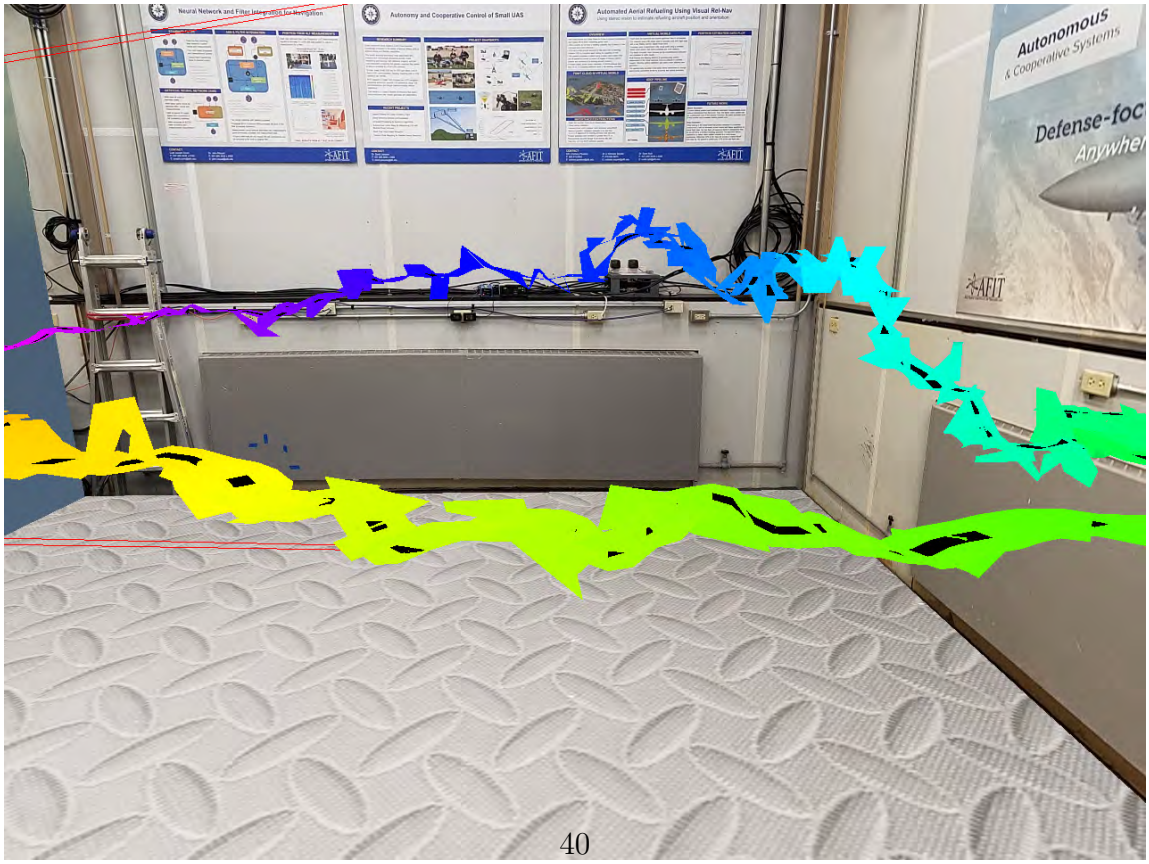


Figure 23. Histogram of Z Error

that time. The top image is the error created using real world imagery, while the bottom shows the error using synthetic imagery.



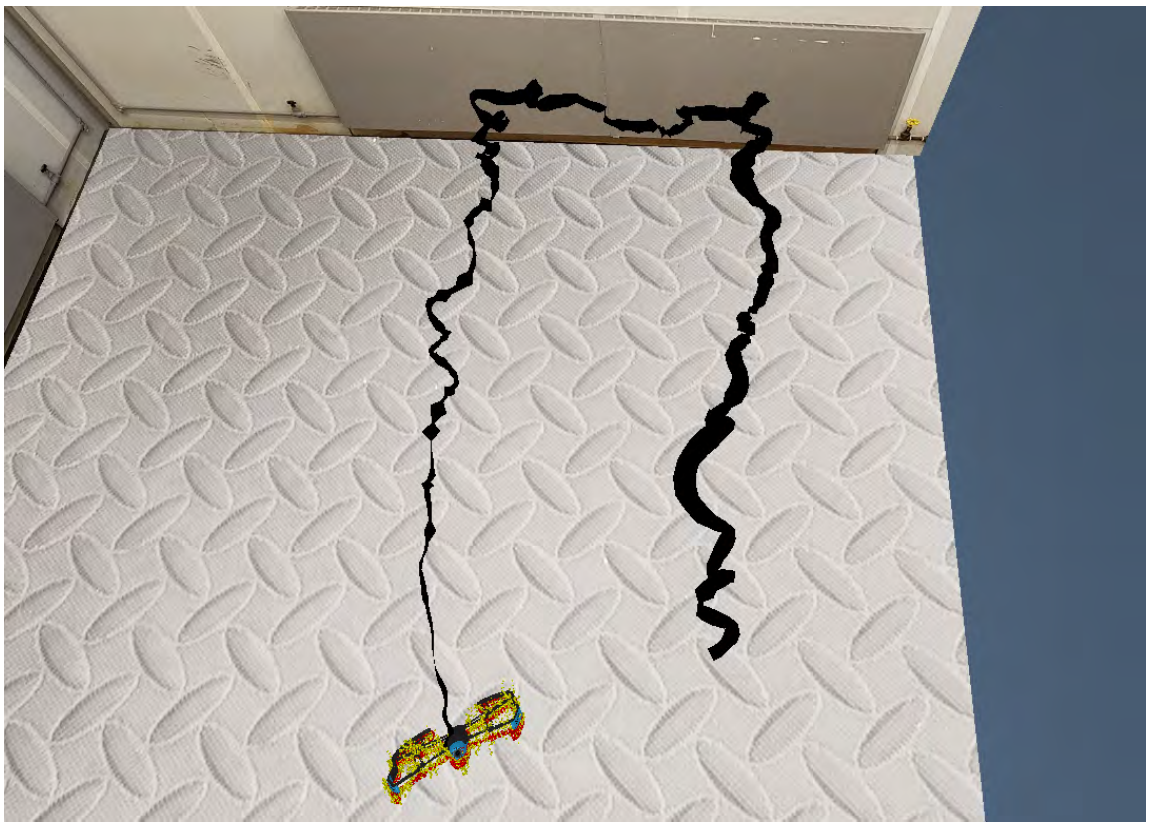
(a) Using Real World Imagery



(b) Using Synthetic Imagery

Figure 24. Truth vs Estimated Position Visualized

The trends in the data can easily be seen when visualized. By taking the magnitude of the error, a ribbon can represent the distance between the truth position and the estimated position. Figure 25 shows the magnitude of the error, where the width of the ribbon is the error in cm. Behavioral trends can be seen when the drone is centered in the envelope and when the drone is on the far right of the envelope. The error magnitude produced on the right is the largest section regardless of the image source. Minimal error is seen in the center of both experiments. Generally, this is due to image distortion. The edges and corners of images are skewed causing more mismatched points. Image distortion is unavoidable regardless of environment, but is minimized through camera calibration.



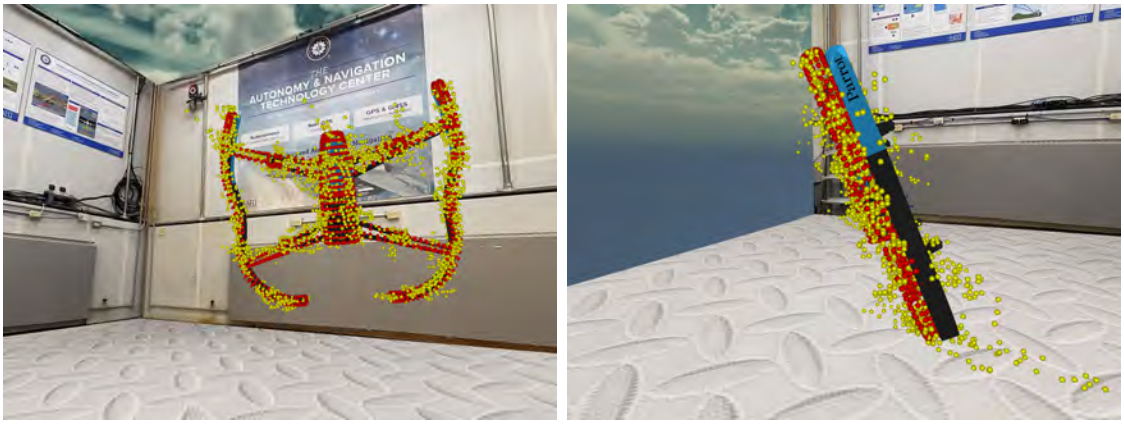
(a) Using Synthetic Imagery

(b) Using Real World Imagery

Figure 25. Magnitude of the Error Visualized

4.3 Reasons for Errors

There are several contributing factors as to why ICP produces estimation error. One reason is the sensed point cloud lies on the top of the drone. Even with a perfect point cloud, a full modeled reference point cloud will have an offset and sit in front of the drone as seen in Figure 26. Despite the tight point cloud, the side view of the drone shows the reference model with the offset. This error can be reduced by shelling the model similar to [35]. If only half of the reference model is used, a successful match will sit closer to the drone center point.



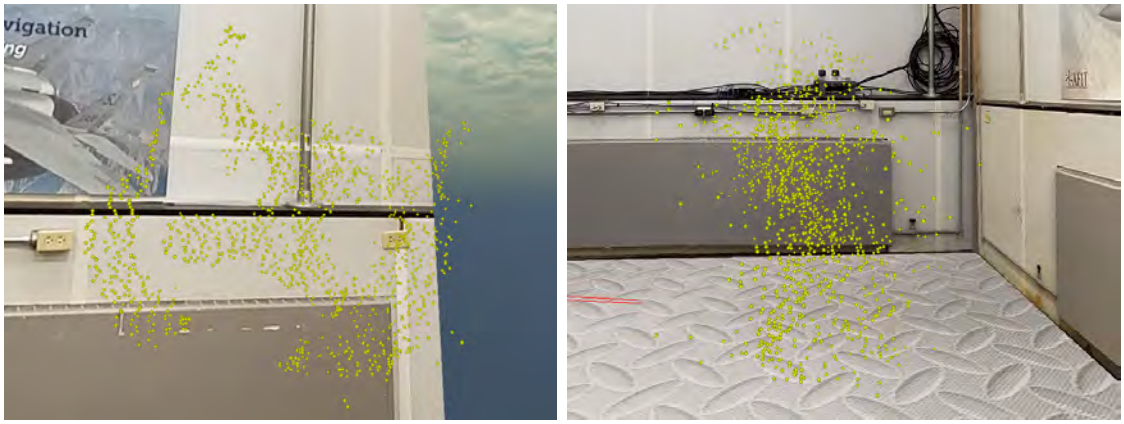
(a) Front View **(b) Side View**

Figure 26. Offset of the Reference Point Cloud

Computer vision is known to produce less sensed points for objects that are farther away. Farther distance also creates a point cloud that is dispersed in the X direction, or the depth from the camera. Figures 27 and 28 show a point cloud generated by real world imagery. Figure 27 represents a point cloud 2 meters from the cameras while Figure 28 represents a point cloud at 4.2 meters from the cameras. It is clear that at a closer range the points are more numerous and tightly packed. This is due to the close baseline of the cameras. There are less features in the image pairs the greater the distance. Less features equates to a smaller point cloud. It is for this reason the estimation error is greater in the X direction than the Y or Z directions.

(a) Front View **(b) Side View**

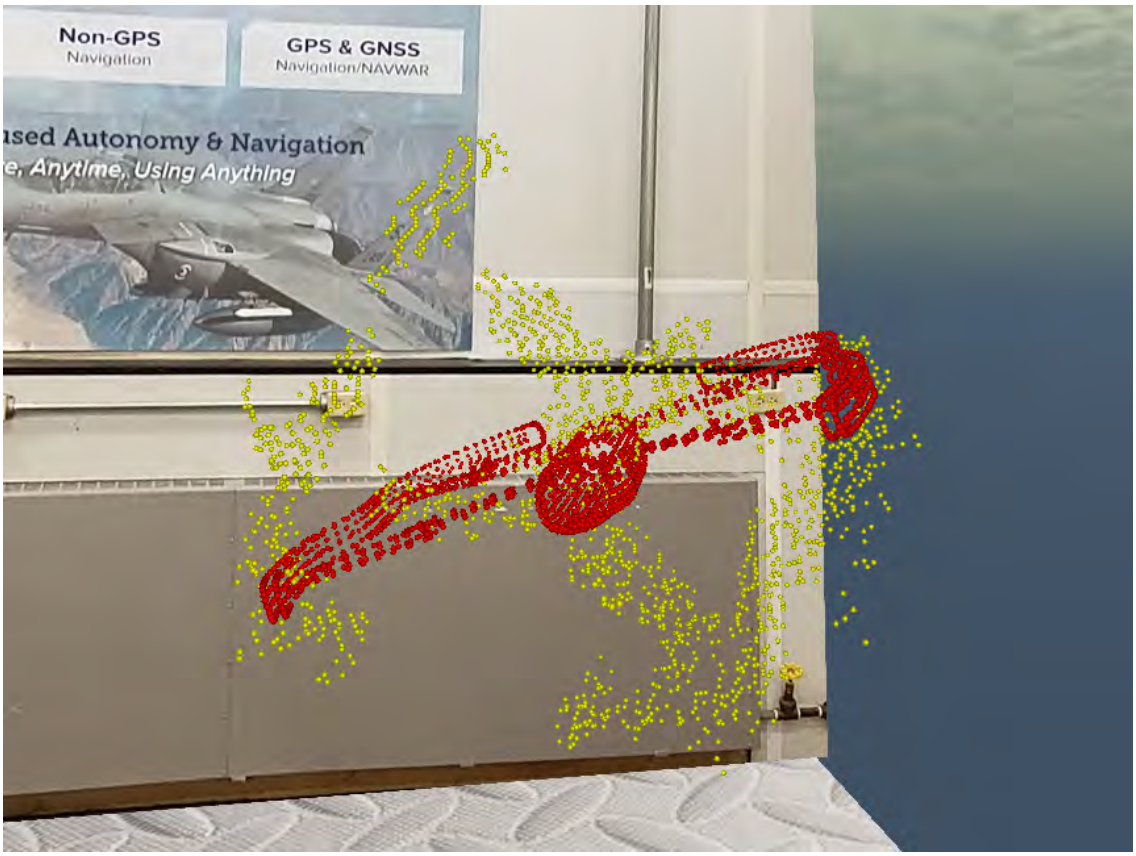
Figure 27. Sensed Point Cloud at 2m



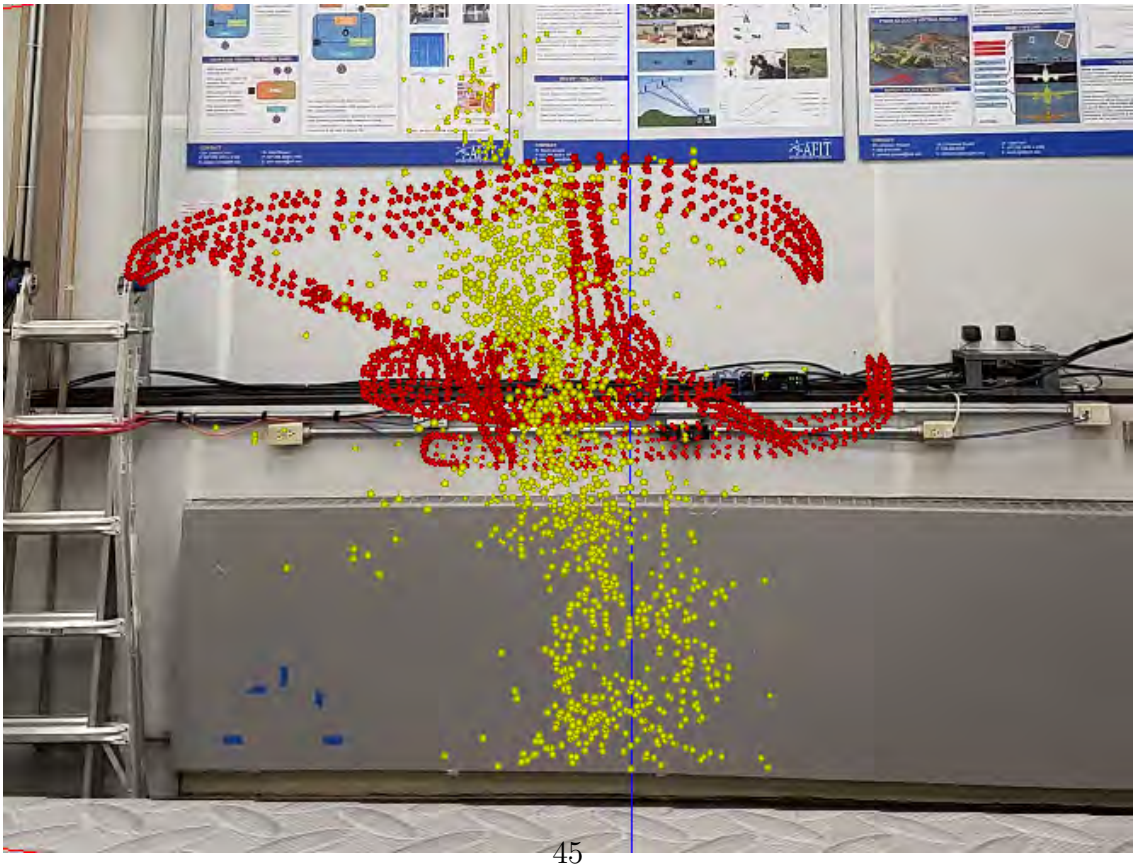
(a) Front View **(b) Side View**

Figure 28. Sensed Point Cloud at 4.2m

The quality of the point cloud is not the only factor that can cause errors. The orientation of the drone proved difficult to estimate. While the spacing of the point cloud caused orientation errors, the shape of the drone did not prove to be ideal. This shape has many opportunities to get stuck in a local minimum as seen in Figure 29. The symmetrical properties of the drone as well as the cross sections create a poor model. An aircraft with wings and a tail would provide a better 3D model to match with.



(a) Front View



(b) Side View

Figure 29. Poor Registration

V. Conclusion

5.1 State of AAR

Due to the latency between the operator and the UAV, it is not possible to perform a mid flight refueling mission on a UAV. The stereo vision system on a modern tanker provides potential for computer vision algorithms. This system can be used to sense and guide the UAV in the refueling envelope. This approach saves time and money by using hardware already installed on the tanker. This method does not require adding sensors to the receiver. Experiments performed in the 3D virtual world and the real world have shown promising results toward this method.

5.2 Research Conclusions

Experiments were performed in the real world and recreated in the virtual world. The real world experiments were captured using physical cameras while the virtual experiments were captured using synthetic sensors. The real and synthetic imagery went through the same computer vision pipeline. Using ICP, the drone's position estimation was produced. The real and synthetic imagery produced position estimation errors within 0.6% of each other. Additionally, similar behavioral trends produced by computer vision and ICP were examined. It was found that these behaviors existed in both real and synthetic imagery solutions.

5.3 Research Contributions

This research shows the similarities of computer vision when using real and synthetic imagery. These similarities are close enough to assume the virtual world can be used as a substitute to the real world in preliminary tests. Capturing virtual

experiments with synthetic data greatly reduces the hassle involved when planning, executing and paying for real world tests.

5.4 Recommendations for Future Work

This work was done in a small scale environment. The imagery collected was also using color images. Other cameras can sense infrared wavelengths and may produce a different result with computer vision. For this reason, data from a real world flight experiment should be used and duplicated in the virtual world. Computer vision can then be performed on this experiment to evaluate the virtual and real environments.

Drone swarms are a current interest at AFIT. It may be possible to estimate the 6DoF of multiple objects. This would allow for positions to be measured on board each drone. This would greatly reduce the communication traffic throughout the swarm.

This research did not provide accurate estimations for the orientation of the drone. This was due to the drone's odd shape. It would be easier for ICP to estimate the orientation of an object such as an airplane. This can further demonstrate the value of using a virtual world for the AAR problem.

Bibliography

1. G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

2. Yoram Yekutieli, Rea Mitelman, Binyamin Hochner, and Tamar Flash. Analyzing octopus movements using three-dimensional reconstruction. *Journal of Neurophysiology*, 98(3):1775–1790, 2007.

3. Richard K Smith. *Seventy-five years of inflight refueling*. Air Force History and Museums Program, 1998.

4. USAF. Usaf aerial refueling. Accessed: http://www.airforce-technology.com/features/featurethe-worlds-best-aerial-refuelling-aircraft/ on 25 January, 2018.

5. USN. Navy aerial refueling. Accessed: https://www.military.com/defensetech/2015/04/22/navy-conducts-first-aerial-refueling-of-x-47b-carrier-launched-drone on 25 January, 2018.

6. Simon Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, 2012.

7. Stuart Russel and Peter Norvig. *Atrificial Intelligence: a Modern Approach*. Pearson, 2009.

8. Richard Szeliski. *Computer Vision: Algorithms and Applications (Texts in Computer Science)*. Springer, 2011.

9. Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

10. Zhengyou Zhang. A Flexible New Technique for Camera Calibration (Technical Report). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2002.

11. Paul Besl and Neil McKay. A Method for Registration of 3-D Shapes, 1992.

12. Jiaolong Yang, Hongdong Li, and Yunde Jia. Go-ICP: Solving 3D registration efficiently and globally optimally. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1457–1464, 2013.

13. Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

14. Xiao Zhang, Craig Glennie, and Arpan Kusari. LiDAR Using a Weighted Anisotropic Iterative Closest Point Algorithm. 8(7):3338–3346, 2015.

15. Szymon Rusinkiewicz. Efficient Variants of the ICP Algorithm a r c Levoy. 2001.

16. D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The Trimmed Iterative Closest Point algorithm. *Object recognition supported by user interaction for service robots*, 3(c):0–3, 2002.

17. Gary K. L. Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C. Langbein, Yonghuai Liu, David Marshall, Ralph R. Martin, Xian-Fang Sun, and Paul L. Rosin. Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, 2013.

18. Jace Robinson, Matt Piekenbrock, Lee Burchett, Scott Nykl, Brian Woolley, and Andrew Terzuoli. *Parallelized Iterative Closest Point for Autonomous Aerial Refueling*, pages 593–602. Springer International Publishing, Cham, 2016.

19. Ryan P Dibley and Michael J Allen. Autonomous Airborne Refueling Demonstration Phase I Flight-Test Results. *AIAA Atmospheric Flight Mechanics Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences*, pages 1–19, 2006.

20. Steven M. Ross. Formation flight control for aerial refueling. Master's thesis, AFIT, 2006.

21. Kevin Liu, Christopher Moore, Robert Buchler, Phil Bruner, Alex Fax, Jacob Hinchman, Ba Nguyen, David Nelson, Fred Ventrone, and Brian Thorward. Precision relative navigation solution for autonomous operations in close proximity. *Record - IEEE PLANS, Position Location and Navigation Symposium*, pages 1246–1251, 2008.

22. Marco Mammarella, Giampiero Campa, Marcello R Napolitano, and Brad Seanor. GPS / MV based Aerial Refueling for UAVs. *AIAA Guidance, Navigation and Control Conference and Exhibit*, pages 1–16, 2008.

23. Marco Mammarella, Giampiero Campa, Marcello R. Napolitano, Mario L. Fravolini, Yu Gu, and Mario G. Perhinschi. Machine vision/GPS integration using EKF for the UAV aerial refueling problem. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 38(6):791–801, 2008.

24. Walton R. Williamson, Gregory J. Glenn, Vu T. Dang, Jason L. Speyer, Stephen M. Stecko, and John M. Takacs. Sensor Fusion Applied to Autonomous Aerial Refueling. *Journal of Guidance, Control, and Dynamics*, 32(1):262–275, 2009.

25. Ml Fravolini and Marco Mammarella. Machine Vision Algorithms for Autonomous Aerial Refueling for UAVs Using the USAF Refueling Boom Method. *Innovations in Defence*, pages 95–138, 2010.

26. Joseph A II Curro. Automated aerial refueling position estimation using a scanning LiDAR. Master's thesis, AFIT, 2012.

27. John Valasek, Kiran Gunnam, Jennifer Kimmett, Monish D Tandale, John L Junkins, and Declan Hughes. Vision-based sensor and navigation system for autonomous air refueling. *Journal of Guidance Control and Dynamics*, 28(5):979–989, 2005.

28. M L Fravolini, V Brunori, A Ficola, M La Cava, and G Campa. Feature Matching Algorithms for Machine Vision Based Autonomous Aerial Refueling. *Analysis*, (Mv), 2007.

29. Mario Luca Fravolini, Giampiero Campa, Marcello Napolitano, Antonio Ficola, Information Engineering, and Giampiero Campa. Evaluation of Machine Vision Algorithms for Autonomous Aerial Refueling for Unmanned Aerial Vehicles Submitted to : AIAA Journal of Aerospace Computing, Information and Communication Evaluation of Machine Vision Algorithms for Autonomous Aerial Refueling. (April 2005):1–23.

30. Marco Mammarella, Giampiero Campa, Marcello R. Napolitano, and Mario L. Fravolini. Comparison of point matching algorithms for the UAV aerial refueling problem. *Machine Vision and Applications*, 21(3):241–251, 2010.

31. Haibin Duan and Qifu Zhang. Visual Measurement in Simulation Environment for Vision-Based UAV Autonomous Aerial Refueling. *IEEE Transactions on Instrumentation and Measurement*, 64(9):2468–2480, 2015.

32. Yimin Deng, Ning Xian, and Haibin Duan. A binocular vision-based measuring system for UAVs autonomous aerial refueling. *IEEE International Conference on Control and Automation, ICCA*, 2016-July:221–226, 2016.

33. Bradley D. Denby. Air Force Institute of Technology. Master's thesis, 2016.

34. Kyle P. Werner. Air Force Institute of Technology. Master's thesis, Air Force Institute of Technology, 2015.

35. Christopher A. Parsons. Improving Automated Aerial Refueling Stereo Vision Pose Estimation Using a Shelled Reference Model. Master's thesis, Air Force Institute of Technology, 2017.

36. Lorenzo Pollini, Giampiero Campa, Fabrizio Giulietti, and Mario Innocenti. Virtual Simulation Set-Up for UAVs Aerial Refuelling. *AIAA Guidance, Navigation and Control Conference and Exhibit*, pages 1–8, 2003.

37. Ba T Nguyen and Lt Tong Lin. The Use of Flight Simulation and Flight Testing in the Automated Aerial Refueling Program. *Aerospace*, pages 1–6, 2005.

38. R. Scott Burns, Curtis S. Clark, and Ron Ewart. The Automated Aerial Refueling Simulation at the AVTAS Laboratory. *AIAA Modeling and Simulation Technologies Conference and Exhibit*, pages 1–12, 2005.

39. R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, Oct 2011.

40. Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew

Davison, and Andrew Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 559–568, New York, NY, USA, 2011. ACM.

41. M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 746–753, May 2017.

42. H. Hattori, V. N. Boddeti, K. Kitani, and T. Kanade. Learning scene-specific pedestrian detectors without real data. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3819–3827, June 2015.

43. J. Marn, D. Vzquez, D. Gernimo, and A. M. Lpez. Learning appearance in virtual scenarios for pedestrian detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 137–144, June 2010.

44. A. Filipowicz, J Liu, and A Kornhauser. Learning to Recognize Distance to Stop Siggns Using the Virtual World of Grand Theft Auto 5. In *Transportation Reasearch Board, 96th Annual Meetingl*, 2017.

45. Pierre Merriaux, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. A study of vicon system positioning performance. *Sensors*, 17(7):1591, 2017.

46. S. Nykl, C. Mourning, M. Leitch, D. Chelberg, T. Franklin, and Chang Liu. An Overview of the STEAMiE Educational Game Engine. In *Frontiers in Education Conference, 2008. FIE 2008. 38th Annual*, pages F3B–21–F3B–25, Oct 2008.

47. Scott Nykl, Chad Mourning, Nikhil Ghandi, and David Chelberg. A Flight Tested Wake Turbulence Aware Altimeter. In George Bebis, editor, *Advances in Visual Computing*, volume 6939 of *Lecture Notes in Computer Science*, pages 219–228. Springer International Publishing, 2011.

48. Zachary C. Paulson. Mitigating the Effects of Boom Occlusion on Automated Aerial Refueling Through Shadow Volumes. Master's thesis, Air Force Institute of Technology, 2018.

49. Thomas R. Stuart. Integrity Monitoring for Automated Aerial Refueling. Master's thesis, Air Force Institute of Technology, 2018.

50. Claire C Gordon, Cynthia L Blackwell, Bruce Bradtmiller, Joseph L Parham, Patricia Barrientos, Stephen P Paquette, Brian D Corner, Jeremy M Carson, Joseph C Venezia, Belva M Rockwell, Michael Mucher, and Shirley Kristensen. 2012 anthropometric survey of U.S. Army Personnel: Methods and summary statistics, 2014.

51. J Bouguet. Camera Calibration Toolbox for Matlab. Accessed: http://www.vision.caltech.edu/bouguetj/calib_doc/ on 12 October, 2017.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
03–23–2018	Master's Thesis	Sept 2016 — Mar 2018

4. TITLE AND SUBTITLE

Stereo Vision: A Comparison of Synthetic Imagery vs Real World Imagery for the Automated Aerial Refueling Problem

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Seydel, Nicholas J., 2LT, USAF

5d. PROJECT NUMBER

16G189

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-18-M-059

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Ba T Nguyen
Aerospace Systems Directorate, Air Force Research Laboratory
2210 8TH ST
WPAFB OH 45433-7765
(937) 938-7765
Email: ba.nguyen@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RQ

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Missions using unmanned aerial vehicles (UAV) have increased in the past decade. However, there is not currently a way to refuel these aircraft. Accomplishing automated aerial refueling (AAR) can be made possible using the stereo vision system on the KC-46. Real world experiments for the AAR problem are expensive and a logistical nightmare. Currently, research performed in a virtual world has shown promising results using computer vision. It is possible to use the virtual world as a substitute environment for the real world. This research compares the performance of stereo vision algorithms on synthetic and real world imagery.

15. SUBJECT TERMS

Automated Aerial Refueling, Virtual World, Computer Vision, Position Estimation, Relative Navigation, Vicon, Synthetic Imagery

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Scott Nykl, PhD, AFIT/ENG
U	U	U	U	66	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4395; scott.nykl@afit.edu